



**Escuela Superior  
de Ingeniería y Tecnología**  
Universidad de La Laguna

## Trabajo de Fin de Grado

---

Aplicación móvil para la generación y  
preparación de menús saludables  
personalizados

*Mobile application for the generation and preparation of  
personalized healthy menus*

Óscar Jesús Socas González

---

La Laguna, 5 de julio de 2019

**Dª. Gara Miranda Valladares**, con N.I.F. 78.563.584-T profesora Contratada Doctor adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

**D. Eduardo Manuel Segredo González**, con N.I.F. 78.564.242-Z profesor Ayudante Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como co-tutor

## **C E R T I F I C A**

Que la presente memoria titulada:

*"Aplicación móvil para la generación y preparación de menús saludables personalizados"*

ha sido realizada bajo su dirección por D. **Óscar Jesús Socas González**, con N.I.F. 43.832.444-X.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 5 de julio de 2019

## Agradecimientos

En primer lugar, quisiera agradecer a mi tutora Gara Miranda Valladares, por guiarme en el desarrollo de este proyecto, por todo el apoyo y la ayuda prestada y por estar siempre con una sonrisa dispuesta a atenderme.

En segundo lugar, me gustaría agradecer a todos los profesores de la carrera, que me han aportado grandes conocimientos, no solo para afrontar este TFG, sino para poder afrontar cualquier proyecto futuro que se me presente.

También quería agradecer a mi pareja, ya que gracias a su apoyo y los ánimos que me ha proporcionado en todo momento he podido finalizar este proyecto.

No puedo olvidarme de las grandes amistades que he hecho en la carrera, ya que junto a ellos y ellas, las horas estudiando en la biblioteca se hacen más llevaderas y cuando tienes algún problema o tropiezo siempre están para ayudarte.

Por último, me gustaría agradecer a mis padres y hermana por siempre creer en mí y apoyarme en todas mis decisiones.

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

## **Resumen**

*En los últimos años ha tenido lugar un cambio radical en el tipo de alimentación que lleva la sociedad. Esto se debe a que en las últimas décadas se ha producido un aumento drástico en el consumo de grasas y alimentos procesados, reduciendo así el consumo de otros alimentos con mayor aportación nutricional. También en los últimos años, se ha generado un aumento en el consumo tecnológico, tanto es así que en la actualidad cuesta encontrar una persona que no disponga de un smartphone, tablet o similar. Es por ello que sería idóneo dotar al usuario de una herramienta tecnológica que genere de forma automática menús saludables, controlando los requisitos nutricionales personales, recomendados por entidades especializadas en nutrición.*

*Para conseguir esto, se ha desarrollado una aplicación móvil para el sistema operativo Android, haciendo uso del lenguaje de programación Java y el entorno de desarrollo Android Studio. La aplicación consta de un generador de menús semanales automatizado, adaptándose a las necesidades fisiológicas del usuario, controlando el consumo calórico semanal, así como los distintos macronutrientes que este debe consumir (proteínas, hidratos y grasas), ajustándose a los márgenes de los distintos grupos alimenticios, asegurando una gran variedad de recetas y adaptándose a distintos estilos de vida o alérgenos que este posea. También dispone de una lista de la compra y despensa interconectadas entre sí, con los distintos ingredientes necesarios para la elaboración del menú.*

*Con el desarrollo de este proyecto se ha conseguido generar una herramienta accesible y simple de usar, que facilite la planificación semanal de las distintas comidas, asegurando una alimentación equilibrada y variada, dejando al usuario libre de esta carga.*

**Palabras clave:** generador de menús, nutrición, aplicación móvil, menú semanal.

## **Abstract**

*In recent years, there has been a radical change in the type of food consumed by society, this is because in recent decades there has been a drastic increase in the consumption of fats and processed foods, reducing the consumption of other foods with more nutrients. Also in recent years, it has increased the technological consumption, so much so that at present it is difficult to find a person who does not have a smart phone, tablet or similar. For these reasons, it would be appropriate to give the user a technological tool that automatically generates healthy menus, controlling the personal nutritional requirements, recommended by entities specialized in nutrition.*

*To achieve this, I've created a mobile application for the Android operating system, using the Java programming language and the Android Studio development environment. The application has an automated weekly menu generator, adapting to the physiological user needs, controlling the weekly caloric consumption, also the different macronutrients that it must consume (proteins, carbohydrates and fats), adjusting to the margins of the different food groups, having a wide variety of recipes and adapting to different lifestyles or allergens that it has. It also has a shopping list and pantry interconnected with each other, with the different ingredients necessary for the preparation of the menu.*

*With the development of this project it has been possible to create an accessible and simple to use tool, that facilitates the weekly planning of the different meals ensuring a balanced and varied diet, leaving the user free to do this work.*

**Keywords:** menu generator, nutrition, mobile application, weekly menu.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	1
1.2. Planificación . . . . .	2
<b>2. Dietética y nutrición: Estado del arte</b>	<b>4</b>
2.1. Pautas y recomendaciones alimenticias . . . . .	4
2.1.1. Energía . . . . .	4
2.1.2. Macronutrientes . . . . .	5
2.1.3. Grupos alimenticios . . . . .	6
2.2. Aplicaciones y herramientas . . . . .	6
2.2.1. Recetas (Eroski Consumer) . . . . .	6
2.2.2. Lifesum . . . . .	7
2.2.3. Mi Menú Semanal . . . . .	8
2.2.4. Hatcook . . . . .	9
2.2.5. Mealime . . . . .	10
2.2.6. Objetivo bienestar . . . . .	11
2.2.7. Cocina para hombres . . . . .	12
2.2.8. Food planner . . . . .	13
2.2.9. Eat this much . . . . .	14
2.2.10. Menú planner . . . . .	15
2.2.11. ¿Qué hay en la nevera? . . . . .	16
2.2.12. Consideraciones generales . . . . .	17
2.3. Bases de datos y API's de recetas . . . . .	19
2.3.1. Bedca.net . . . . .	19
2.3.2. Nutritionix . . . . .	19
2.3.3. WebKnox Recipe API . . . . .	19
2.3.4. EDAMAM . . . . .	19
2.3.5. Dietopro . . . . .	19
2.3.6. Kiwilión . . . . .	20
2.4. API seleccionada . . . . .	20
<b>3. Tecnologías para el desarrollo de aplicaciones móviles</b>	<b>21</b>
3.1. Android Studio . . . . .	21
3.2. Java . . . . .	22
3.3. JavaScript . . . . .	22
3.3.1. Appcelerator Titanium . . . . .	22
3.4. HTML5 + CSS . . . . .	22
3.5. Kotlin . . . . .	23
3.6. Basic 4 Android (B4A) . . . . .	23

3.7. Mono for Android . . . . .	23
3.8. LiveCode . . . . .	23
3.9. Ruboto . . . . .	24
3.10 Selección de la tecnología para el desarrollo de la aplicación . . . . .	24
<b>4. MealTime: Casos de uso</b>	<b>25</b>
4.1. Modelo de datos . . . . .	26
4.2. Gestión de usuarios . . . . .	27
4.3. Características de los menús . . . . .	28
4.4. Gestión de la compra y la despensa . . . . .	34
4.5. Búsquedas de recetas e información nutricional . . . . .	38
4.6. Idioma . . . . .	40
4.7. Repositorio . . . . .	41
<b>5. Conclusiones y líneas de trabajo futuras</b>	<b>42</b>
5.1. Conclusiones . . . . .	42
5.2. Líneas de trabajo futuro . . . . .	43
<b>6. Conclusions and Future work</b>	<b>44</b>
6.1. Conclusions . . . . .	44
6.2. Future work . . . . .	44
<b>7. Presupuesto</b>	<b>46</b>

# Índice de Figuras

2.1. Recetas (Eroski Consumer) . . . . .	6
2.2. Lifesum . . . . .	8
2.3. Mi Menú Semanal . . . . .	8
2.4. Hatcook . . . . .	9
2.5. Mealime . . . . .	10
2.6. Objetivo bienestar . . . . .	11
2.7. Cocina para hombres . . . . .	12
2.8. Food planner . . . . .	13
2.9. Eat this much . . . . .	14
2.10Menú planner . . . . .	15
2.11¿Qué hay en la nevera? . . . . .	16
4.1. Icono de la aplicación MealTime . . . . .	25
4.2. Pantalla de bienvenida de MealTime . . . . .	27
4.3. Pantallas del cuestionario del usuario . . . . .	28
4.4. Menú desplegable . . . . .	28
4.5. Pantalla de inicio . . . . .	29
4.6. Pantalla del menú semanal . . . . .	30
4.7. Pantalla del menú semanal con un día finalizado . . . . .	30
4.8. Código de conexión con la API EDAMAME . . . . .	31
4.9. Pantalla del almuerzo de un día de la semana . . . . .	32
4.10Pantalla de información del plato seleccionado . . . . .	33
4.11Pantalla de preparación del plato . . . . .	33
4.12Pantalla de la tabla nutricional del plato . . . . .	34
4.13Esquema del funcionamiento de MealTime . . . . .	34
4.14Pantalla de la despensa del usuario . . . . .	35
4.15Pantalla de la lista de la compra . . . . .	36
4.16Ingrediente que se encuentra en la despensa . . . . .	36
4.17Ingrediente que se encuentra en la despensa con poca cantidad . . . . .	37
4.18Clase Adapter . . . . .	37
4.19Método onClick . . . . .	38
4.20Pantalla de búsqueda . . . . .	39
4.21Pantalla de información nutricional del usuario . . . . .	39
4.22Fichero strings.xml . . . . .	40
4.23Selección de un nuevo idioma . . . . .	41

# Índice de Tablas

1.1. Planificación de tareas. . . . .	3
2.1. Cálculo del gasto energético de reposo en varones y mujeres. . . . .	4
2.2. Estimación del gasto energético para distintos niveles de actividad física. .	5
2.3. Frecuencia de consumo por grupos de alimentos. . . . .	6
2.4. Tabla resumen de las herramientas. . . . .	18
4.1. Tabla <code>datosUsuario</code> en SQLite. . . . .	26
4.2. Tabla <code>macronutrientes</code> en SQLite. . . . .	26
4.3. Tabla <code>platos</code> en SQLite. . . . .	26
4.4. Tabla <code>lista_ingredientes</code> en SQLite. . . . .	26
4.5. Tabla <code>despensa</code> en SQLite. . . . .	27
4.6. Tabla <code>datos_menu</code> en SQLite. . . . .	27
7.1. Presupuesto de las tecnologías usadas . . . . .	47
7.2. Presupuesto del coste por tarea . . . . .	47

# **Capítulo 1**

## **Introducción**

Uno de los principales problemas que atañe a la sociedad actual es la forma inadecuada de alimentarse que sufre parte de la población. La nutrición es la clave para tener una buena salud, tener energía y controlar el peso de manera natural y sin riesgos. Pero, debido a las exigencias de nuestras actividades cotidianas y a la falta de conocimiento en temas de nutrición, resulta difícil lograr una buena, balanceada y completa alimentación. Es por esto, que en las últimas décadas ha aumentado de forma considerable el consumo de las denominadas comidas rápidas y snacks, las cuales se alejan mucho de las necesidades nutricionales de cada individuo. El principal problema es que la mayoría de estos alimentos han perdido sus propiedades nutricionales a causa de los distintos procesos de transformación que sufren antes de llegar a consumirse. La ausencia de planificación en las distintas comidas del día, también se debe a la falta de tiempo en la sociedad actual y a un ritmo de vida más ajetreado y rápido, por lo que pueden suponer un reto en la toma de buenas decisiones en cuanto a la dieta y nutrición se refiere. Este problema de malnutrición, unido a una poca frecuencia en la realización de actividad física, ha producido un aumento considerable de diabetes, sobrepeso y obesidad dando lugar a cifras alarmantes no solo en España, sino también a nivel mundial. Para evitar estos problemas es necesario seguir una alimentación que satisfaga las necesidades nutricionales y fisiológicas del individuo.

En los últimos años ha surgido, en las distintas redes sociales, un movimiento denominado “*Real Food*”, el cual se define como un estilo de vida que se basa en comer comida real y evitar productos ultraprocesados, ya que estos son el mayor enemigo de nuestra dieta. Este movimiento lo han fomentado distintos dietistas y nutricionistas por sus redes sociales como son el caso de Gabriela Uriarte, Carlos Ríos, Aitor Sánchez, etc. Debido a la repercusión de este movimiento, da a entender que la población poco a poco se va interesando más en llevar una buena alimentación, por lo que sería interesante proporcionar una herramienta que facilite este proceso a largo plazo [32].

### **1.1. Objetivos**

El objetivo de este proyecto es el análisis, diseño e implementación de una aplicación móvil para la resolución del problema de la planificación de menús. Este consiste en la generación para el usuario de un plan dietético, generalmente semanal, compuesto por platos o menús saludables para las principales comidas del día. El plan alimenticio debe estar basado en estándares nutricionales. Además, es importante que el menú generado

se adapte a las condiciones fisiológicas del usuario, ya que la cantidad de nutrientes recomendada varía según estos factores en particular.

También resulta interesante disponer de un sistema que no se limite a ofrecer planes dietéticos nutricionalmente adecuados, sino que también sea capaz de adaptar dichos planes a las necesidades particulares del usuario como, por ejemplo, sus preferencias personales sobre el tipo de comida, la variedad en los platos, las características estéticas de los alimentos, las incompatibilidades con determinados alimentos por cuestión de alergias, intolerancias o estilos de vida específicos, etc.

Por lo tanto, los objetivos mínimos que se quieren obtener son los siguientes:

- Gestionar características concretas o fisiológicas del usuario como el tipo de actividad física que este realice, etc.
- Disponer de una base de datos de recetas amplia.
- Que en dicha base de datos se tenga información de la preparación de las recetas y los ingredientes para las elaboraciones.
- Disponer de información nutricional de cada plato.
- Crear un generador de menús automatizado según las características del usuario y las recomendaciones realizadas por entidades especializadas en nutrición.
- Proporcionar una lista de la compra y una despensa, para que se puedan organizar los distintos ingredientes.

## 1.2. Planificación

Con el fin de abordar el problema de la generación de menús saludables, se propusieron una serie de tareas o actividades que se relacionan en la Tabla 1.1 y que han servido de guía para la planificación de este trabajo:

	<b>Objetivo</b>	<b>Descripción</b>
<b>ACT1</b>	Análisis del estado del arte	Analizar las aplicaciones o herramientas existentes en el ámbito de la dietética y la nutrición, especialmente aquellas que incluyen planificación automática de menús.
<b>ACT2</b>	Identificación de mecanismos para la especificación de menús alimenticios	Realizar una recopilación del tipo de lenguajes o mecanismos que se utilizan a la hora de especificar los platos, alimentos o ingredientes que conforman un menú alimenticio. Hacer una propuesta concreta para utilizar por la herramienta a desarrollar.
<b>ACT3</b>	Identificación de bases de datos alimentarias	Realizar una recopilación de bases de datos sobre platos, recetas, alimentos con sus respectivas propiedades nutricionales, etc.
<b>ACT4</b>	Diseño e implementación de la herramienta	Diseñar e implementar una herramienta informática que proporcione al usuario una planificación automática de menús saludables adaptados a sus necesidades o preferencias, así como los procedimientos para la elaboración de los menús y la adquisición de los alimentos necesarios.
<b>ACT5</b>	Documentación	Elaboración del material de soporte para el uso de la aplicación así como la memoria del TFG.

Tabla 1.1: Planificación de tareas.

# Capítulo 2

## Dietética y nutrición: Estado del arte

### 2.1. Pautas y recomendaciones alimenticias

En este apartado, se explicarán las distintas pautas y recomendaciones que se deben tener en cuenta para la generación de un menú semanal adaptado a las distintas características del usuario.

#### 2.1.1. Energía

Para realizar el cálculo energético en reposo, es necesario saber el sexo del usuario, la edad, el peso y la altura del mismo y una vez sepamos esto, haciendo uso de la Tabla 2.1, realizamos el cálculo en base a la fórmula correspondiente en cada caso [31].

SEXO	EDAD (años)	GER (kcal/día)
Varón	18-30	$15,4 \times \text{peso} - 27 \times \text{altura} + 717$
	31-60	$11,3 \times \text{peso} + 16 \times \text{altura} + 901$
	>60	$8,8 \times \text{peso} + 1,128 \times \text{altura} - 1,071$
Mujer	18-30	$13,3 \times \text{peso} + 334 \times \text{altura} + 35$
	31-60	$8,7 \times \text{peso} - 25 \times \text{altura} + 865$
	>60	$9,2 \times \text{peso} + 637 \times \text{altura} - 302$

Tabla 2.1: Cálculo del gasto energético de reposo en varones y mujeres.

La columna etiquetada como *GER* indica la fórmula a aplicar para calcular el *gasto metabólico en reposo*. De cara a aplicar las fórmulas, el peso debe expresarse en *kg* y la altura en *metros*.

Para realizar una estimación del gasto energético diario según los distintos niveles de actividad física, es necesario multiplicar el gasto metabólico en reposo (*GER*) por un factor de corrección que se obtiene de la Tabla 2.2 y que depende del tipo de actividad física que realice el usuario [31].

Categoría de actividad	Factor de corrección
Reposo. Sueño, tumbado	1,0
Sedentaria o ligera. Actividades que se hacen sentado o de pie, como pintar, conducir, trabajo de laboratorio, escribir a máquina, planchar, cocinar, jugar a las cartas, tocar un instrumento, ver la televisión	1,55
Activa o moderadamente activa. Practicar "footing", ciclismo, tenis, bailar y otros deportes aeróbicos	1,75
Muy activa o intensa. Caminar con carga cuesta arriba, cortar árboles, cavar con fuerza, baloncesto, escalada, futbol, rugby	2,10

Tabla 2.2: Estimación del gasto energético para distintos niveles de actividad física.

### 2.1.2. Macronutrientes

Para poder realizar el cálculo de **Lípidos** o grasas diarias, es necesario haber calculado previamente el gasto energético diario. Teniendo esta información, debemos distinguir si el usuario cocina con aceite de oliva o con otro tipo de aceite, ya que estos últimos son más perjudiciales para la salud. Por lo tanto, si el usuario cocina con aceite de oliva, debe consumir del total energético diario una cantidad menor o igual al 35 % y si cocina con otro tipo de aceite distinto al de oliva, debe consumir del total energético una cantidad menor o igual al 30 %. Para poder hacer conversiones de kcal a gramos o al revés, debemos tener en cuenta que un gramo de grasas equivale aproximadamente a 9kcal.

En cuanto a las **proteínas** deberían aportar aproximadamente del 8 al 15 % de las calorías totales ingeridas por la persona. Para el cálculo diario de estas, se debe disponer del peso del usuario, ya que las recomendaciones proteicas actuales de ingestas diarias para la población general, se sitúan en torno a 0,8 gramos de proteína por kg de peso corporal. En caso de que la persona realice ejercicio físico de forma activa, la ingesta apropiada sería entre 1,0 y 1,2 g/día por kg de peso corporal en mujeres y de 1,2 a 1,4 g/día por kg de peso corporal en hombres. En deportistas que llevan a cabo entrenamiento de fuerza, los rangos recomendados oscilan entre 1,2 y 1,7 g/día por kg de peso corporal. Para poder hacer conversiones de kcal a gramos o al revés, debemos tener en cuenta que un gramo de proteína equivale aproximadamente a 4kcal.

No se han establecido necesidades diarias precisas de **hidratos de carbono** en la alimentación humana. Sin embargo, las recomendaciones en los países occidentales aconsejan que este nutriente constituya entre el 55 y 60 % del total de la energía. Para realizar el cálculo de hidratos de carbono, se obtienen primero los lípidos y las proteínas. Lo restante del total energético diario corresponderá a los hidratos de carbono que debe consumir la persona a diario. Para poder hacer conversiones de kcal a gramos o al revés, debemos tener en cuenta que un gramo de hidratos equivale aproximadamente a 4kcal.

### **2.1.3. Grupos alimenticios**

Los principales nutrientes se encuentran distribuidos en los diferentes alimentos; un solo alimento no nos puede aportar todos los nutrientes esenciales. Es por ello que estos se han clasificado en función del nutriente predominante y el consumo habitual recomendado. Estas agrupaciones se denominan grupos alimenticios. En la Tabla 2.3 se muestra una relación de los diferentes grupos de alimentos considerados junto con su frecuencia de consumo recomendada [35]:

<b>Grupo de alimentos</b>	<b>Frecuencia recomendada</b>
Leche y derivados	2-4 raciones/día
Pan, cereales, cereales integrales, arroz, pasta, papas	4-6 raciones/día
Verduras y hortalizas	Al menos 2 raciones/día
Frutas	Al menos 3 raciones/día
Aceite de oliva	3-6 raciones/día
Legumbres	2-4 raciones/semana
Frutos secos	3-7 raciones/semana
Pescados y mariscos	3-4 raciones/semana
Carnes magras, aves	3-4 raciones/semana
Huevos	3-4 raciones/semana

Tabla 2.3: Frecuencia de consumo por grupos de alimentos.

## **2.2. Aplicaciones y herramientas**

En este apartado, hablaremos de las distintas aplicaciones y herramientas que existen en la actualidad para generar menús dietéticos y comentaremos en líneas generales su funcionamiento.

### **2.2.1. Recetas (Eroski Consumer)**



Figura 2.1: Recetas (Eroski Consumer)

Esta aplicación ha sido desarrollada por la cadena de supermercados Eroski / Consumer. Presume de tener miles de recetas las cuales disponen de un informe de enfermedades para las que están recomendadas, todo eso elaborado por expertos en nutrición.

Ventajas:

- Dispone de una lista de ingredientes.
- Indica los pasos para elaborar el plato.
- Proporciona la tabla nutricional del plato.
- Permite controlar alérgenos.
- Permite crear un menú con sus platos.
- Da la posibilidad de crear la lista de la compra, añadir platos a favoritos e indicar si se padece algún tipo de enfermedad.
- Posee un sistema de cálculo de calorías que se deben consumir diariamente según el sexo, edad, altura, peso y actividad física diaria.
- Ajusta los ingredientes al número de comensales que se indiquen.

Desventajas:

- La aplicación en sí es publicidad del supermercado Eroski y te indica el precio aproximado del plato si compras los ingredientes en ese supermercado.
- Muchos de los platos no tienen informe de enfermedades para los que están recomendados, tal y como indican en la descripción de la aplicación.
- Hay platos en los que se contradicen sus recomendaciones con las de la OMS (Organización mundial de la salud) [21], en los cuales, por ejemplo, se recomienda ciertas cantidades de azúcares diarios muy por encima del máximo recomendado por la OMS en una persona sana (la aplicación recomienda 90g por plato y la OMS recomienda como máximo 25g diarios).
- La mayoría de los platos que pude analizar eran insanos.
- No permite controlar estilos alimenticios como el veganismo, vegetarianismo, etc.
- No genera un menú personalizado.
- No genera la lista de la compra.

## 2.2.2. Lifesum

Se trata de una herramienta que permite contar las calorías que se consumen, planificar dietas específicas y compaginarlas con el registro del deporte y las actividades que se realicen al día. También posee un sistema de escáner de códigos de barras para registrar los productos que se consumen de forma rápida.



Figura 2.2: Lifesum

Ventajas:

- Al entrar pregunta datos personales del usuario para poder adaptar las dietas y recomendaciones al individuo, también pregunta si se desea ganar peso, perder peso o simplemente llevar una vida saludable.
- Permite de forma gratuita contabilizar la cantidad de calorías que se consume a diario según los productos que se ingieren, al igual que el máximo de calorías que se deben consumir.
- Da la posibilidad de registrar el deporte que se hace, con wearables de forma automática o indicando el tipo de ejercicio realizado y el tiempo haciéndolo.
- Permite añadir gran cantidad de productos por medio del lector de código de barras.

Desventajas:

- Todas las recetas, menús dietéticos y elaboraciones son de pago por lo que no pudimos comprobar cuan buenas o no eran estas.
- Hay productos que el lector de código de barras no detecta y los que sí detecta, hay mucha información que para acceder debes pagar cierta cantidad.
- Por lo que pude investigar en la parte gratuita de la aplicación, no controla estilos alimenticios como el veganismo, vegetarianismo, etc. Tampoco controla alérgenos.

### 2.2.3. Mi Menú Semanal



Figura 2.3: Mi Menú Semanal

Se trata de una herramienta gratuita para organizar el menú semanal y la lista de la compra en función del menú creado.

Ventajas:

- Permite añadir tus propias recetas o usar las que aparecen en el “Recetario” de la app, creadas por otros usuarios.
- Dispone de un lector de recetas para poder escuchar la preparación del plato.
- En cuanto a la variedad de recetas, dispone de 58 platos.
- Se indica la receta para la preparación del plato, la lista de ingredientes y una imagen del plato.
- Permite al usuario crearse un menú semanal.
- Proporciona la lista de la compra del menú semanal que se haya creado.

Desventajas:

- Tiene publicidad en la parte inferior.
- No es una aplicación intuitiva.
- Dispone de platos que son saludables y otros que no lo son.
- No se genera un menú semanal automático, lo tiene que crear el usuario.
- No controla estilos alimenticios como el veganismo, vegetarianismo, etc y tampoco controla alérgenos.

#### 2.2.4. Hatcook



Figura 2.4: Hatcook

Se trata de una aplicación de recetas de cocina gratuita, en la cual puedes encontrar recetas de todo tipo y compartir tus propias recetas con el resto de usuarios. Presume de tener más de 9.000 recetas.

Ventajas:

- Realiza el cálculo automático de los ingredientes según la cantidad de comensales.
- Posee de un modo de lectura para escuchar la elaboración sin tener que mirar la pantalla.

- Con el modo off line se puede acceder a la app sin necesidad de conexión a internet.
- La app es sencilla, intuitiva y estéticamente muy llamativa.
- En cada plato nos aparece una imagen o video, el nombre, el autor, los ingredientes y la elaboración.
- Separa los platos por colecciones como caseras, fáciles, económicas, desayunos, saludables, etc.
- Dispone de una sección de categorías donde se agrupan los platos por ingredientes, ya sean arroces, verduras, pasta y pizzas, legumbres, etc.
- Es gratuita y sin publicidad.

Desventajas:

- No genera menús de forma automática ni te da la posibilidad de crearlos tú mismo.
- No permite controlar estilos alimenticios como el veganismo, vegetarianismo, etc.
- No controla alérgenos.
- No genera la lista de la compra.

## 2.2.5. Mealime



Figura 2.5: Mealime

*Mealime* es una aplicación para planificar el menú semanal con los platos saludables de la app, al entrar pregunta qué tipo de dieta desea seguir, si una clásica sin restricciones, una con pocos carbohidratos, una vegetariana, una vegana, etc. Por último, pregunta por la cantidad de porciones, según la cuantía que se suela comer o la cantidad de comensales.

Ventajas:

- Los platos que proporciona la app son muy variados y saludables.
- Controla alérgenos.
- Al entrar por primera vez te pregunta si hay algún ingrediente que no te gusta.

- Indica si un plato es de elaboración compleja.
- Permite realizar un menú semanal.
- Genera la lista de la compra de forma automática y permite eliminar productos de la lista que ya tengas en casa.
- En cada plato indica los utensilios que se necesitarán, los ingredientes y las instrucciones de preparación del plato.

Desventajas:

- El menú semanal lo has de realizar tú, no se genera automáticamente.
- Para ver la tabla nutricional debes pagar cierta cantidad anual o mensual.

## 2.2.6. Objetivo bienestar



Figura 2.6: Objetivo bienestar

*Objetivo bienestar* [20] es una página web que proporciona información en muchos sectores como alimentación, salud, mente, belleza, deportes, etc. Nos centraremos en la parte de alimentación de la misma, para ser más concreto en la sección de menú semanal saludable. En ella se nos muestra una serie de recomendaciones de las cantidades que se debe consumir de ciertos productos semanalmente. Al final de dicha página se proporciona una serie de publicaciones con menús y platos con distintos temas como: “menú semanal de recetas rápidas y sencillas”, “menú semanal contra la retención de líquidos”, “menú para comer sano y de táper para toda la semana”, “menú semanal saludable y ligero”, etc.

Ventajas:

- Contiene menús veganos, vegetarianos y sin lactosa.
- Proporciona varios menús semanales.
- Dispone de artículos con algunos consejos de alimentación.
- La mayoría de los platos son saludables.

Desventajas:

- Los menús no son personalizados.
- No controla alérgenos.

- Al seleccionar uno de estos menús, nos muestra el menú semanal, normalmente con 5 comidas al día y en alguno de los platos te indica la preparación y los ingredientes del mismo, pero en otros los pasa por alto, tal vez por la sencillez del plato.
- La tabla nutricional no aparece para todos los platos.
- No proporciona la lista de la compra para cada menú semanal.

### 2.2.7. Cocina para hombres



Figura 2.7: Cocina para hombres

*Cocina para hombres* [17] es una página web para generar menús semanales, al entrar te permite seleccionar las recetas según la estación del año. Por defecto están todas las estaciones seleccionadas. Posteriormente para el primer plato, te da a elegir cuantos días deseas comer a la semana ciertos platos, es decir del 1 al 7 cuantos días quieras comer ensaladas, frituras, empanadillas y tostadas, etc y lo mismo con el segundo plato. Por último, se genera el menú.

#### Primeros platos

Indica cuántos días de cada tipo de plato quieras.

Ensaladas	Frituras, empanadillas y tostadas	Potajes y Sopas	Arroz, legumbres, patatas y pasta	Verduras, champiñones y setas
<input type="button" value="3 ⇄"/>	<input type="button" value="0 ⇄"/>	<input type="button" value="3 ⇄"/>	<input type="button" value="2 ⇄"/>	<input type="button" value="1 ⇄"/>

#### Segundos platos

Indica cuántos días de cada tipo de plato quieras.

Huevos	Pescados y Mariscos	Carnes	Aves	Caza
<input type="button" value="1 ⇄"/>	<input type="button" value="2 ⇄"/>	<input type="button" value="2 ⇄"/>	<input type="button" value="1 ⇄"/>	<input type="button" value="1 ⇄"/>

Ventajas:

- En cada plato te indica los ingredientes y la receta para elaborarlo.
- El menú se genera de forma automática.
- Es una herramienta gratuita.
- Según los datos que le introduzcas al inicio genera diferentes menús.

Desventajas:

- No se genera un menú saludable, ya que hay platos que no lo son.
- No controla hábitos de vida como el ser vegano, vegetariano, etc.
- No tiene en cuenta alérgenos.
- No genera la lista de la compra.
- No muestra imagen del plato.
- La estética es simple y poco llamativa.

### 2.2.8. Food planner



Figura 2.8: Food planner

*Food planner* es una aplicación que permite planear las comidas diarias y semanales, usando una selección de recetas descargadas de webs populares y blogs de cocina. También posee planes semanales o de varias semanas ya creados.

Ventajas:

- Permite importar gran cantidad de platos externos a la aplicación.
- En el inventario puedes añadir los productos que se tienen en casa y también da la opción de hacerlo por medio de un lector de código de barras.
- Posee una lista de la compra.
- Permite crear varios menús semanales con los platos importados.

Desventajas:

- Posee mucha publicidad, que ocupa la pantalla completa y mucha de ella con sonido.
- El lector de código de barras no reconoce la mayoría de los productos (en las pruebas realizadas no reconoció ningún producto), lo que si permite añadirlo y la próxima vez que se escanee lo reconozca.

- Los platos no se encuentran en la app, se tienen que importar o crear de forma manual.
- El menú no se genera automáticamente, se debe crear manualmente con los platos importados.
- Muchos de los platos importados no tienen ingredientes ni tabla nutricional, esto es debido a que poseen distintos formatos o incluso algunos son solo videos de cómo preparar el plato.
- La lista de la compra no se genera automáticamente, tienes que ir plato por plato e indicar que se añadan los ingredientes de dicho plato a la lista, pero esto no siempre funciona bien, debido a que hay platos que no tienen registrados los ingredientes.
- Posee una sección llamada "Tienda FP" en la cual hay distintos menús dietéticos o planes alimenticios, con distintas características, pero todos son de pago.
- Se trata de una aplicación muy poco intuitiva en la cual el usuario se puede perder con facilidad.
- No controla hábitos de vida como el ser vegano, vegetariano, etc.
- No controla alérgenos.

## 2.2.9. Eat this much



Figura 2.9: Eat this much

Se trata de una aplicación móvil que planifica automáticamente las distintas comidas según una serie de datos que le pide al usuario al ingresar por primera vez en la herramienta, ajustándose a una serie de calorías que calcula de forma automática.

Ventajas:

- Tiene en cuenta datos del usuario como la altura, el peso, la meta u objetivos a conseguir, la constitución corporal, la edad y el sexo para generar los platos.
- Controla hábitos de vida como el veganismo, el vegetarianismo, etc, también tiene en cuenta distintos alérgenos y pregunta si hay algo en específico que no le guste al usuario comer.

- Con los datos recogidos, hace un cálculo de la cantidad diaria que debes consumir de calorías, carbohidratos, grasas y proteínas. También permite modificar los rangos de dichos datos, al igual que limitar la cantidad de sodio y de colesterol en los planes.
- Genera un menú diario automáticamente y también permite crearlo manualmente con los platos que tiene la app. Dicho menú se ajusta a la cantidad de calorías, carbohidratos, grasas y proteínas que se indicaron.
- El menú se puede modificar cambiando algunos platos por otros o generando un nuevo menú diferente.
- En cada plato muestra los ingredientes, la receta para prepararlo y un precio estimado en dólares.
- En cada plato permite indicar si te gusta o no el plato, añadirlo a favoritos o si quieres que ese alimento se repita con cierta frecuencia en los menús que se generen a continuación.
- Dispone de una lista con todos los platos y otra con todos los ingredientes, por si se quiere añadir alguno de estos a los platos o al menú.

Desventajas:

- No todos los platos son saludables, hay algunos que utilizan productos insanos como un exceso de mayonesa o papas fritas de bolsa, es decir no sigue una dieta mediterránea.
- Solo permite generar el menú del día actual, si se desea planificar la semana entera es necesario pagar por una cuenta premium.
- No genera una tabla nutricional completa ya que solo controla calorías, carbohidratos, grasas y proteínas.
- No genera una lista de la compra del menú creado.

## 2.2.10. Menú planner



Figura 2.10: Menú planner

*Menú planner* [19] es una página web publicitada por la empresa Nestlé en la cual se generan tanto menús semanales, como menú diario o simplemente la cena del día. Cada semana se genera automáticamente un menú distinto.

Ventajas:

- Dispone de un menú semanal.

- Cada plato contiene la lista de ingredientes, la preparación y la tabla nutricional.
- Algunos de los platos tiene un video explicando cómo prepararlos.
- Permite crear una cena para el día actual preguntando que es lo que se ha comido ese día y si se quiere generar un único plato o un primer y segundo plato, con esos datos genera una lista de platos que puedes seleccionar para cenar.

Desventajas:

- Los menús son genéricos, es decir, no pide datos personales para adaptarse a tus necesidades.
- Muchos de los platos los elabora con productos Nestlé siempre indicando su marca en los ingredientes.
- No todas las recetas son sanas y muchos de los menús semanales tiene un día en el que te recomienda comer chocolates y bizcochos.
- No controla hábitos de vida como el ser vegano o vegetariano.
- No controla alérgenos.
- El menú semanal no se puede modificar, es estático e igual para todo el mundo.
- No genera la lista de la compra.

### **2.2.11. ¿Qué hay en la nevera?**



Figura 2.11: ¿Qué hay en la nevera?

¿Qué hay en la nevera? [25] es una página web en la que los usuarios comparten conocimientos por medio de comentarios, puntuando los platos o creando recetas. También consta de un generador de menú tanto semanal como único, es decir platos para una comida puntual.

Ventajas:

- Para crear el menú, realiza una serie de preguntas como, por ejemplo, escoger los tipos de comidas que se quiere preparar en esa semana (arroces, carnes, pescados y mariscos, ensaladas y verduras, guisos, pastas, pizzas y pasteles y por último bocadillos). Da a elegir para los primeros platos entre sopas, cremas y entrantes. También pregunta si quieres cocinar postres y si también se van a generar cenas. Por último, permite indicar si eres vegetariano, si solo quieres recetas light, para llevar en tupper o solo cosas rápidas y sencillas.
- Genera de forma automática un menú semanal con almuerzo y cena.

- Permite generar otro menú por si no te gusta el que se creó.
- Muestra los ingredientes, los pasos a seguir para preparar el plato, el tiempo estimado y la dificultad del mismo.

Desventajas:

- Algunos de los platos no tiene imágenes.
- Para la generación del menú, no pregunta datos como la edad, peso, sexo o la constitución física.
- Controla el ser vegetariano, pero no otros hábitos de vida como el veganismo.
- No controla alérgenos.
- No genera la lista de la compra para el menú semanal.

### **2.2.12. Consideraciones generales**

Se ha decidido crear una tabla resumen de todas las aplicaciones y herramientas analizadas teniendo en cuenta ciertas características generales, las cuales se quieren abordar en este TFG.

Cabe destacar que, durante el análisis, se han encontrado más herramientas relacionadas con la dietética y nutrición, pero debido a que no aportaban funcionalidades nuevas a las herramientas ya analizadas, solo se han añadido en la tabla resumen que veremos a continuación (Tabla 2.4).

Como se puede observar en la Tabla 2.4 ninguna de las herramientas analizadas sobre dietética y nutrición cumplen con los objetivos de este TFG, es por ello que vamos a desarrollar nuestra aplicación móvil para la generación y preparación de menús saludables personalizados.

	<b>Plataforma</b>	<b>Receta</b>	<b>Ingredientes</b>	<b>Menú</b>	<b>Menú automatizado</b>	<b>Alérgenos</b>	<b>Hábitos de vida</b>	<b>Lista de la compra</b>	<b>Datos del usuario</b>	<b>Gratis</b>
<b>Recetas (Eroski Consumer)</b>	App	Sí	Sí	Sí	No	Sí	No	No	No	Sí
<b>Lifesum</b>	App	-	-	Sí	No	-	No	-	Sí	No
<b>Mi Menú Semanal</b>	App	Sí	Sí	Sí	No	No	No	Sí	No	Sí
<b>Hatcook</b>	App	Sí	Sí	No	No	No	No	Sí	No	Sí
<b>Mealime</b>	App	Sí	Sí	Sí	No	No	No	Sí	No	Sí, con parte de pago
<b>Objetivo bienestar</b>	Web	A veces	A veces	Sí	No	No	Sí	No	No	No
<b>Cocina para hombres</b>	Web	Sí	Sí	Sí	Sí	No	No	No	No	Sí
<b>Food plan-ner</b>	App	A veces	A veces	Sí	No	No	No	Sí	No	Sí, con parte de pago
<b>Eat this much</b>	App	Sí	Sí	Sí	Sí, solo del día actual	Sí	No	Sí	Sí	Sí, con parte de pago
<b>Menú plan-ner</b>	Web	Sí	Sí	Sí	No	No	No	No	No	Sí
<b>¿Qué hay en la nevera?</b>	Web	Sí	Sí	Sí	Sí	No	Solo ser vegetariano	No	No	Sí
<b>Nooddle</b>	App	Sí	Sí	No	No	Sí	Sí	No	Sí	Sí
<b>Webosfritos</b>	Web	Sí	Sí	Sí	No	No	No	No	No	Sí
<b>Tesco Real food</b>	Web	Sí	Sí	Sí	No	Sí	No	No	No	Sí

Tabla 2.4: Tabla resumen de las herramientas.

## **2.3. Bases de datos y API's de recetas**

Para poder desarrollar la aplicación y generar un menú, es necesario disponer de una base de datos o API que proporcione recetas con los ingredientes necesarios y la preparación de las mismas. Por este motivo se llevó a cabo una búsqueda exhaustiva a través de la plataforma *RapidAPI* [22], este es un sitio web que contiene una gran variedad de APIs de todo tipo.

### **2.3.1. Bedca.net**

*Bedca* [15] es una red de Centros de investigación públicos, Administración e Instituciones privadas, cuyo objetivo es el desarrollo y mantenimiento de la Base de Datos Española de Composición de Alimentos. Posee una gran lista de alimentos con información muy detallada de cada uno de ellos.

### **2.3.2. Nutritionix**

*Nutritionix* [11] presume de ser la base de datos de nutrición más grande del mundo, con más de 5 millones de consultas realizadas al mes por los usuarios de esta, con muchísima información de distintos restaurantes, comidas y alimentos. Esta resulta ser de pago por lo que no se pudo probar.

### **2.3.3. WebKnox Recipe API**

La API de recetas de *WebKnox* [24] permite acceder a más de 330.000 recetas. Da la posibilidad a los usuarios de buscar recetas en lenguaje natural como "brownies sin azúcar". Esta API también contiene mucha información sobre ingredientes, nutrición, precios, consejos, etc. Además, dispone de varios *widgets* para visualizar listas de ingredientes, precios, etc. El principal inconveniente es que es de pago.

### **2.3.4. EDAMAM**

*EDAMAM* [18] es una API que dispone de versión gratuita con más de 1.5 millones de recetas accesibles a los usuarios, con nutrición detallada de cada una de ellas, permite filtros por distintos estilos alimenticios, alérgenos o tipo de dieta (baja en grasas, alta en proteína, etc).

### **2.3.5. Dietopro**

*Dietopro* [5] es un software de gestión dietético nutricional con *sistema experto autómata* que permite crear un plan dietético personalizado, editarlo, imprimirla e incluso enviarla. Cuenta con una base de datos de 3543 platos diferentes y permite la exportación de los mismos en distintos formatos. La principal desventaja de esta herramienta es que es exclusiva para titulaciones oficiales en dietética y nutrición.

### 2.3.6. Kiwilimón

*Kiwilimón* [16] es una aplicación tanto web como móvil, que proporciona gran cantidad de recetas con sus ingredientes, preparación, puntuación de otros usuarios, etc. También dispone de una API de recetas tanto en español como en inglés, que promete ser muy interesante. Para acceder a dicha API era necesario solicitar permiso e indicar para qué se usaría la misma. Les escribí un correo indicando que se usaría para un Trabajo de Fin de Grado, del Grado en Ingeniería Informática, pero nunca recibí respuesta.

## 2.4. API seleccionada

Entre las API's y bases de datos analizadas que me proporcionaran recetas, he decidido hacer uso de la API *EDAMAM* [18] debido a que esta es gratuita y tiene ciertas cualidades que nombraremos a continuación:

- Permite la búsqueda de recetas, con sus ingredientes y preparación.
- Dispone de una tabla nutricional muy completa para cada plato, con más de 25 nutrientes.
- Permite filtrar por tipo de dieta (vegetariana, vegana, etc) y por alérgenos.
- Tiene versión en español y en inglés, aunque la versión española está en beta.
- Tiene una gran variedad de platos, aseguran tener más de 1.5 millones de recetas.

Para poder realizar búsquedas de recetas en la API *EDAMAM*, primero debes solicitar un *id* y una *clave*, luego debes usar la ruta <https://test-es.edamam.com/search> para obtener las recetas en español (esta versión está actualmente en beta) o la ruta <https://api.edamam.com/search> para obtener las recetas en inglés. La construcción de la ruta para solicitar una búsqueda es la siguiente:

```
https://api.edamam.com/search?q={palabra_a_buscar}&app_id=${tu_id}&app_key=${tu_clave}
```

También puedes indicar más parámetros en la url para filtrar la búsqueda, un ejemplo sería el siguiente:

```
https://api.edamam.com/search?q=chicken&app_id=f8a801f5&app_key=70a3be2482007e101a7617c95f9438425a&from=0&to=20&health=alcohol-free
```

En este caso se realiza una búsqueda de la palabra *pollo*, indicando el *id* y la *clave*, luego le señalamos que se quiere hacer una búsqueda de las primeras 20 recetas que aparezcan y que estas sean sin alcohol. La respuesta que proporciona la API se da en formato *JSON*.

# Capítulo 3

## Tecnologías para el desarrollo de aplicaciones móviles

En este capítulo se hablará de las distintas tecnologías que he encontrado para programación móvil, con una pequeña descripción de las mismas. Por último, se indicará la que he seleccionado para la implementación del proyecto, con los motivos por los que se ha elegido.

### 3.1. Android Studio

*Android Studio* [1] es el entorno de desarrollo integrado (*IDE*) oficial para el desarrollo de aplicaciones para *Android*, está basado en el software *IntelliJ IDEA* (entorno de desarrollo integrado para el desarrollo de programas) y ha sido publicado de forma gratuita a través de la Licencia *Apache 2.0*. Posee un potente editor de códigos y herramientas para desarrolladores de *IntelliJ*. Ha sido diseñado específicamente para el desarrollo de *Android*.

Las funciones principales de *Android studio* son:

- Sistema de compilación basado en *Gradle* flexible.
- Emulador rápido con varias funciones.
- Entorno unificado en el que puedes realizar desarrollos para todos los dispositivos *Android*.
- *Instant Run* para aplicar cambios mientras la app se ejecuta sin la necesidad de compilar un nuevo *APK*.
- Integración de plantillas de código y *GitHub* para ayudar a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y *frameworks* de prueba.
- Herramientas *Lint* para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con *C++*, *Java*, *Kotlin* y *NDK*.
- Soporte incorporado para *Google Cloud Platform*, lo que facilita la integración de *Google Cloud Messaging* y *App Engine*.

## 3.2. Java

Java [29] es el lenguaje de programación nativo que usa *Android*, por lo tanto, cualquier aplicación que use directamente el *hardware* y se comunique con el sistema operativo, usará este código. Es un lenguaje muy extendido, presente en muchos dispositivos, tanto por la red como a nivel local, no solo permite programar para *Android*, también nos permite desarrollar programas de escritorio, o incluso videojuegos.

Java es un lenguaje de programación que mucha gente usa para el desarrollo de aplicaciones en *Android*, y junto con *Android Studio* es la pareja de lenguaje y entorno de desarrollo recomendada por *Google* para el desarrollo de dichas aplicaciones.

## 3.3. JavaScript

*JavaScript* [7] es un lenguaje de programación interpretado, definido como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se trata de un lenguaje web el cual junto a plataformas como *Appcelerator Titanium* [2] permite compilar de manera nativa para poder programar en dispositivos móviles. Es más sencillo y ligero que programar en el lenguaje nativo. El procedimiento sería el siguiente, primero se escriben las instrucciones en *JavaScript* y luego *Appcelerator Titanium* se encarga de traducirlo al lenguaje nativo, no sólo de *Android*, también de *iOS* o *Windows 10 mobile*. De modo que se podrá acceder, también, a los recursos físicos del dispositivo como la cámara, el GPS, el acelerómetro, etc.

### 3.3.1. Appcelerator Titanium

*Titanium* [2] es una plataforma creada por la empresa *Appcelerator* que permite desarrollar aplicaciones para dispositivos móviles (*iOS*, *Android* y *Blackberry*) programando en *Javascript*. Desde el ambiente de desarrollo de *Titanium* se crea la interfaz gráfica y se programa el comportamiento en *javascript*, y en base a esto, el motor de *Titanium* genera un proyecto nativo en *Xcode* (en el caso de *iOS*) o un proyecto nativo de *Android*. Con esto, se puede compilar utilizando las herramientas correspondientes para generar ejecutables nativos para cada plataforma. Empresas de la talla de Avis, Zipcar, Adidas e*Bay* o *PayPal* emplean esta herramienta para sus aplicaciones móviles.

También existen herramientas similares a esta como por ejemplo *PhoneGap* [13] que es un paquete de librerías que permite empaquetar aplicaciones *HTML5* [6], *CSS* [4] y *JavaScript* [13] de manera que puedan ser usadas como apps para móviles o *Web Apps*.

## 3.4. HTML5 + CSS

*HTML* [6] es un lenguaje de marcado que se utiliza para el desarrollo de páginas web. *CSS* [4] es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Ambos son lenguajes para el desarrollo web, fueron creados para internet, pero el paso del tiempo los ha beneficiado, sobre todo con la llegada de la versión 5 de *HTML*. Esta versión ofrece muchas posibilidades respecto la versión 4, entre ellas reproducir vídeo de un modo

más nativo en el navegador, sin la ayuda de *flash*, algo que reduce considerablemente el consumo de recursos. Al igual que ocurre con *JavaScript*, plataformas como *Titanium* permiten el desarrollo de este tipo de aplicaciones con estos lenguajes. El concepto es el mismo que el usado para el desarrollo web, una estructura web que, en vez de apuntar a pantallas grandes de escritorio, apunta a pantallas más pequeñas de dispositivos móviles.

El resultado de las apps que se crean con esta herramienta quedan lejos de poder compararse con aplicaciones nativas, el resultado se aproxima más a una web adaptada a *smartphones*.

### 3.5. Kotlin

*Kotlin* [8] es un lenguaje de programación de tipado estático que corre sobre la máquina virtual de *Java* y que también puede ser compilado a código fuente de *JavaScript*. Aunque no tiene una sintaxis compatible con *Java*, *Kotlin* está diseñado para interoperar con código *Java* y es dependiente del código *Java* de su biblioteca de clases. Dispone del soporte oficial de *Google* y está inspirado en *Java*, *C++* y todos sus precedentes. Es un lenguaje muy limpio y relativamente simple, con menos formalidades y reglas que *C++* y *Java*, pero no se encuentra estandarizado.

### 3.6. Basic 4 Android (B4A)

*Basic4Android* [3] es un entorno comercial que nos permitirá desarrollar aplicaciones para *Android* programando en un lenguaje muy similar a *Visual Basic*, sin embargo en el fondo aún seguirá siendo *Java*. Este entorno no es gratuito ya que hay que pagar una cuota para poder usarlo. *Basic4Android* es una de las grandes enemigas de *Android Studio*. Esta herramienta permite usar cómodamente ciertas librerías que facilitarán el trabajo. Algunas de estas librerías permitirán trabajar con el GPS del móvil, el bluetooth, interacción con sitios web usando HTTP, tratamiento multimedia con archivos locales y streaming o incluso controlando la cámara del móvil, entre otros.

### 3.7. Mono for Android

*Mono for Android* [10] es un componente que permite el desarrollo de aplicaciones para *Android* utilizando *C#* y *Visual Studio*. *Mono for Android*, desarrollado por *Xamarin*, permite crear, depurar y empaquetar una aplicación en un *.apk* y utilizarla directamente en un teléfono *Android*. A diferencia de otras herramientas, *Mono for Android* permite el desarrollo de aplicaciones nativas, significando que se integra al ecosistema *Android* e interactúa con las aplicaciones nativas creadas en *Java*, utiliza la interfaz (UI) nativa del sistema operativo y sigue el modelo de desarrollo de *Android*.

### 3.8. LiveCode

*LiveCode* [9] es una herramienta que permite el desarrollo de aplicaciones multiplataforma, incluyendo un completo constructor de interfaces gráficas de usuario, así

como las herramientas que son necesarias para producir aplicaciones. Sus características principales son:

- Permite exportar aplicaciones para entornos *desktop* (*Windows, Linux y Mac*) y móviles (*Android* y *iOS*).
- Función de "*Drag & Drop*" de los elementos en la ventana.
- Interfaz gráfica sencilla y fácil de interpretar.
- Existe una versión gratuita y varias de pago.
- No usa declaración de tipos. Ej: enteros, boolean, reales etc.

### 3.9. Ruboto

*Ruboto* [14] es un *framework* y herramienta para desarrollar aplicaciones *Android* nativas, utilizando el lenguaje de programación *Ruby*. No es tan grande como *Appcelerator Titanium*, pero aun así tiene una gran cantidad de programadores y también existe bastante documentación para comenzar a programar.

### 3.10. Selección de la tecnología para el desarrollo de la aplicación

De todas las tecnologías analizadas se ha elegido trabajar con **Java** [29] y **Android Studio** [1], por lo que el sistema operativo que se utilizará será *Android*. Los motivos por los que se ha hecho dicha elección se enumeran a continuación:

- *Android Studio* es el entorno de desarrollo integrado (*IDE*) oficial para la elaboración de aplicaciones en *Android* y ha sido diseñado específicamente para el desarrollo de dicho sistema operativo.
- *Java* es el lenguaje de programación nativo que usa *Android*, por lo tanto, cualquier aplicación que use directamente el hardware y se comunique con el sistema operativo, usará este código. Se trata de un lenguaje muy extendido presente en muchos dispositivos y muy demandado.
- Tanto *Android Studio* como *Java* son gratuitos.
- *Java* junto con *Android Studio* son la pareja de lenguaje y entorno de desarrollo recomendada por *Google* para el desarrollo de dichas aplicaciones.

# Capítulo 4

## MealTime: Casos de uso

Para abordar el problema de la generación de menús, se ha creado una aplicación móvil para la gestión y elaboración de un menú semanal, adaptado al usuario según sus características fisiológicas. Esta recibe el nombre de *MealTime* y genera el almuerzo para los siete días de la semana, con su primer plato, segundo y postre. No solo genera menú semanal, también proporciona la visualización de los distintos platos con sus ingredientes, preparación y tabla nutricional, al igual que dispone de una lista de la compra y despensa que están interconectadas entre sí para dar una mejor experiencia de uso.

El motivo principal del desarrollo de esta aplicación es facilitar a las personas la planificación del menú semanal, dejando a estas libres de tener que organizar la lista de la compra o decidiendo los distintos platos que deben consumir para tener una dieta equilibrada. También se quiso que la interfaz fuera simple e intuitiva para que cualquier persona, independientemente de su nivel de conocimientos y manejo de aplicaciones informáticas pudiera usarla. El logo de la aplicación es el que se muestra en la Figura 4.1, este refleja la esencia de la herramienta.



Figura 4.1: Icono de la aplicación MealTime

Lo primero que se hizo fue aprender a programar en *Android Studio* [33] con Java [29][28], ya que nunca antes los había usado, para ello realicé un curso online en La Universidad Politécnica de Valencia por medio de la plataforma edX [30] de introducción a la programación en *Android*.

## 4.1. Modelo de datos

Para hacer uso de toda la información en las distintas pantallas y que esta no se borre al cerrar la aplicación, es necesaria almacenarla en algún sitio. Por ello, para guardar los datos del usuario, la información de los distintos platos, ingredientes, lista de la compra, despensa y demás, se ha hecho uso del sistema de gestión de bases de datos **SQLite**.

Con el fin de almacenar y distribuir de forma correcta toda esta información, se han creado 7 tablas que explicaremos a continuación:

En la tabla **datosUsuario** (Tabla 4.1) se almacena un id para identificar al usuario, el nombre, sexo, edad, altura, la cantidad de actividad que realiza normalmente, el tipo de dieta que sigue y si cocina o no con aceite de oliva.

<b>id</b>	<b>nombre</b>	<b>sexo</b>	<b>edad</b>	<b>altura</b>	<b>peso</b>	<b>actividad</b>	<b>health</b>	<b>aceiteOliva</b>
-----------	---------------	-------------	-------------	---------------	-------------	------------------	---------------	--------------------

Tabla 4.1: Tabla **datosUsuario** en SQLite.

En la tabla **macronutrientes** (Tabla 4.2) se almacena un id para identificar el usuario al que pertenecen los macronutrientes, también se guardan las kilocalorías, grasas, proteínas e hidratos que debe consumir a diario.

<b>id</b>	<b>kcal_dia</b>	<b>grasas</b>	<b>proteinas</b>	<b>hidratos</b>
-----------	-----------------	---------------	------------------	-----------------

Tabla 4.2: Tabla **macronutrientes** en SQLite.

En la tabla **platos** (Tabla 4.3) se almacena un id para identificar el plato, el día de la semana al que pertenece este, el nombre, foto, la URL de la preparación, la URL de la tabla de nutrientes, la cantidad de raciones del plato, sus kilocalorías, grasas, hidratos y proteínas.

<b>id</b>	<b>dia</b>	<b>nombre</b>	<b>foto</b>	<b>url</b>	<b>shareAs</b>	<b>raciones</b>	<b>kcal</b>	<b>grasas</b>	<b>hidratos</b>	<b>proteinas</b>
-----------	------------	---------------	-------------	------------	----------------	-----------------	-------------	---------------	-----------------	------------------

Tabla 4.3: Tabla **platos** en SQLite.

En la tabla **lista\_ingredientes** (Tabla 4.4) se almacena un id para identificar el ingrediente, el **id\_plato** es una clave ajena al id del plato al que pertenece en la tabla **platos**, esto es así para poder identificar que un conjunto de ingredientes pertenecen a un plato en concreto, por ejemplo todos los que tengan el **id\_plato** igual a 5 serán ingredientes del plato con id igual a 5. Luego almacenamos el ingrediente, también se guardan los gramos del mismo y **tiene** se utiliza para indicar si ese alimento se encuentra en la despensa y así saber si mostrarlo en la lista de la compra o no.

<b>id</b>	<b>id_plato</b>	<b>ingrediente</b>	<b>gramos</b>	<b>tiene</b>
-----------	-----------------	--------------------	---------------	--------------

Tabla 4.4: Tabla **lista\_ingredientes** en SQLite.

En la tabla **despensa** (Tabla 4.5) se almacena un id para identificar el ingrediente, también el nombre del mismo y los gramos de este disponibles en la despensa.

<b>id</b>	<b>ingrediente</b>	<b>gramos</b>
-----------	--------------------	---------------

Tabla 4.5: Tabla despensa en SQLite.

En la tabla `datos_menu` (Tabla 4.6) se almacena información relevante de la generación de menús que posteriormente se usará para que el usuario pueda regenerar un plato que no le guste. Se almacena un identificador, la cantidad de platos que llevan legumbres, pescados, carnes y huevos para que no se pase de la dosis recomendada para una semana y por último se almacena el último id que se ha asignado en la tabla `lista_ingredientes`.

<b>id</b>	<b>legumbres</b>	<b>pescados</b>	<b>carnes</b>	<b>huevos</b>	<b>ultimo_id</b>
-----------	------------------	-----------------	---------------	---------------	------------------

Tabla 4.6: Tabla `datos_menu` en SQLite.

## 4.2. Gestión de usuarios

Al acceder a la aplicación aparece una pantalla de bienvenida (Figura 4.2), con el logo de la aplicación y un botón para acceder.



Figura 4.2: Pantalla de bienvenida de MealTime

La primera vez que se entra en la aplicación, solicitará una serie de datos personales al usuario, para posteriormente poder generar el menú, todos estos datos se muestran en la Figura 4.3.

Dicha información se almacena en la base de datos al finalizar el cuestionario, así la próxima vez que se acceda a la herramienta no se tendrán que rellenar de nuevo, simplemente entrará directamente. Desde dentro de la aplicación el usuario podrá modificar todos los datos introducidos, siempre que lo desee, accediendo a los tres puntos que se

Fill in the following personal information:

Name: \_\_\_\_\_

Gender:

- Man
- Woman

Age: \_\_\_\_\_

Height: cm \_\_\_\_\_

Weight: kg \_\_\_\_\_

How active are you on a daily basis?

- Nothing active
- Sedentary, little active
- Active or moderately active
- Very active

Name: \_\_\_\_\_

Gender:

- Man
- Woman

Age: \_\_\_\_\_

Height: cm \_\_\_\_\_

Weight: kg \_\_\_\_\_

How active are you on a daily basis?

- Nothing active
- Sedentary, little active
- Active or moderately active
- Very active

Other characteristics of your diet:

- Vegetarian
- Vegan
- Alcohol-free
- Reduced in sugar
- Without peanuts
- No nuts
- Balanced

For cooking do you use olive oil?

- Yes
- No

**Continue**

**SAVE**

Figura 4.3: Pantallas del cuestionario del usuario

encuentran en la parte superior derecha, ahí aparecerá un menú desplegable (Figura 4.4) y pulsando sobre “*Modify personal data*” podrá cambiar su información personal.

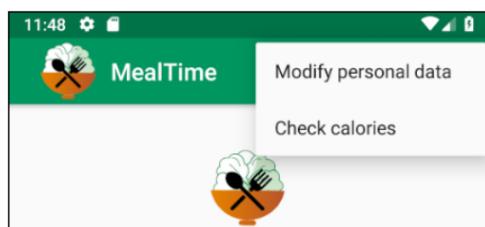


Figura 4.4: Menú desplegable

De los datos que se piden, el nombre se solicita para dar un trato más personalizado al usuario, los datos fisiológicos (sexo, edad, altura y peso) se requieren para realizar el cálculo calórico descrito en la Sección 2.1.1, la pregunta de cuanto de activo es a diario se solicita para poder realizar el cálculo del gasto energético de la Sección 2.1.1. También se solicita el tipo de dieta que realiza, por si la persona es vegetariana, vegana o si es alérgica a algún alimento. Por último, se pregunta si cocina con aceite de oliva o con otro tipo de aceite, ya que esto influirá en el cálculo de grasas del usuario.

### 4.3. Características de los menús

Una vez rellenados los datos anteriores, se accede a la pestaña de inicio (Figura 4.5), en caso de no ser la primera vez que se entra, se accederá directamente a esta pestaña y así se evita introducir los datos una y otra vez. En esta pantalla se muestra un mensaje con el nombre del usuario, preguntando que desea hacer, para ello da una serie de opciones: crear un menú semanal, consultar el menú previamente creado, consultar las calorías

y macronutrientes del usuario, acceder a la lista de la compra, acceder a la despensa o buscar recetas.

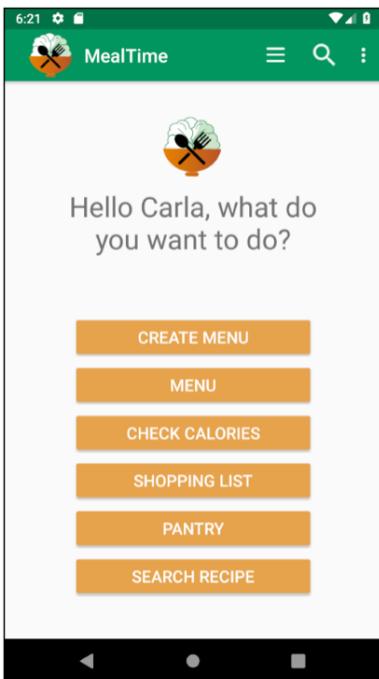


Figura 4.5: Pantalla de inicio

Al pulsar sobre crear un menú, se muestra una animación indicando que se está generando dicho menú, ya que este tarda algo de tiempo, debido a las búsquedas que tiene que hacer. Una vez se ha creado, se abre la pestaña del menú semanal (Figura 4.6), a esta pestaña también se puede acceder pulsando el botón menú de la figura anterior. En esta pestaña se muestran los primeros platos, los segundos y el postre de cada día de la semana, con un botón al lado que al pulsarlo se oscurece ese día indicando que ya se ha realizado (Figura 4.7) y por lo tanto se descontarán los ingredientes usados de la despensa y lista de la compra.

Para poder generar menús es necesario establecer una conexión con la API que va a proporcionar las recetas, en este caso *EDAMAM*. Para ello se hizo uso de un cliente *REST* para *Android* y *Java* llamado *Retrofit* [27], que lo que hace es convertir una petición *HTTP* de la API en una interfaz *Java*. Se añadieron sus dependencias al proyecto y se estableció conexión con la API. En la Figura 4.8 podemos ver un ejemplo de una parte del código donde se establece una conexión con la API y se solicita una lista de platos principales. En la documentación de la herramienta [23] se encuentra toda la información para poder obtener los resultados esperados, un pequeño resumen de esto se explicó en la Sección 2.4.

Los ingredientes que proporciona la API *EDAMAM* se encuentran en una cadena de texto con más información, es decir no aparece solo el ingrediente, por ejemplo “leche”, si no que aparecen con más información alrededor, en el caso de la leche podría ser algo como “un vaso de leche, si es desnatada mejor” y estos no poseen ningún patrón para diferenciarlo del resto de texto. Por ello para conseguir que no se repita el ingrediente principal de un plato dos días seguidos y también para controlar el consumo de los distintos grupos alimenticios, se hizo uso de un diccionario de alimentos [34] para realizar

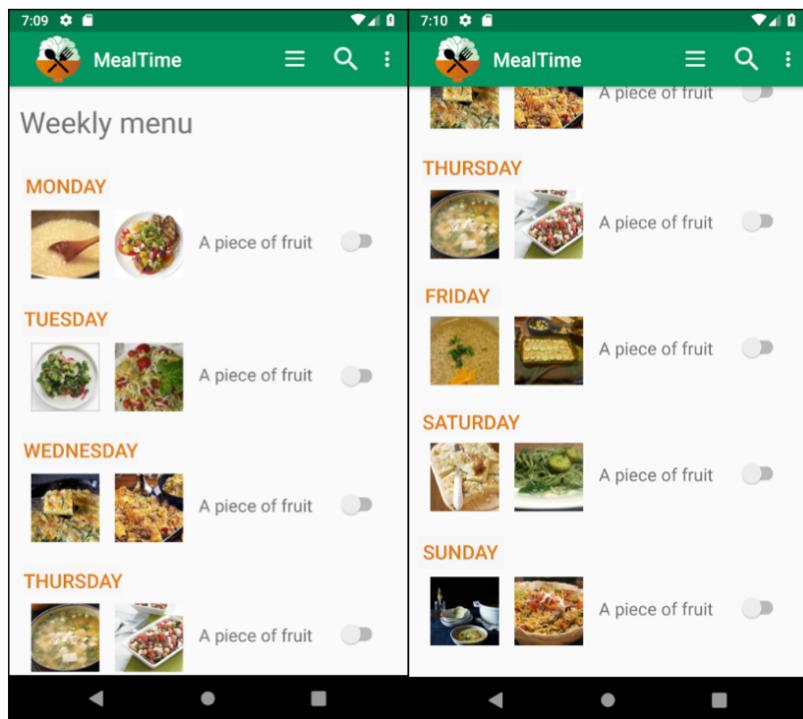


Figura 4.6: Pantalla del menú semanal

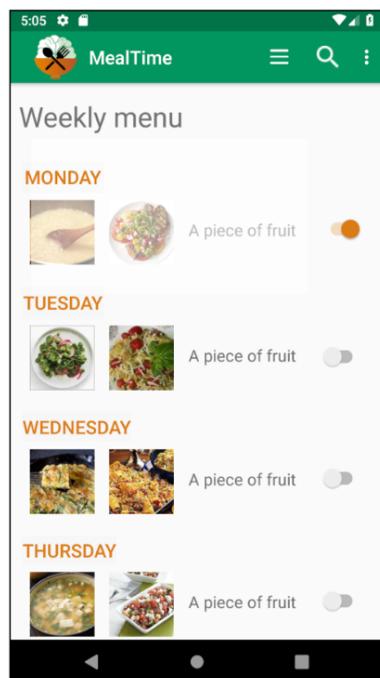


Figura 4.7: Pantalla del menú semanal con un día finalizado

una búsqueda en este, de cada palabra que apareciera en cada ingrediente proporcionado por la API.

Con respecto a la generación del menú semanal, según la OMS [21], aproximadamente el almuerzo debe suponer entre un 35 y un 40 % del consumo total de calorías diarias. Es por ello que para generar el menú debemos ajustarlas a este rango y establecer unas máximas y unas mínimas, ya que al igual que no se puede pasar de las calorías que debe consumir a diario, tampoco se puede quedar muy corto con estas. Para la generación del

```

retrofit = new Retrofit.Builder()
    .baseUrl("https://api.edamam.com/")
    .addConverterFactory(GsonConverterFactory.create())
    .build();

public interface RecipeService {
    @GET("search?from=0")
    Call<Respuesta> obtenerDatos(@Query("to") String to, @Query("q") String q,
                                    @Query("app_id") String app_id, @Query("app_key") String app_key ,
                                    @Query("health") String health, @Query("calories") String calories);
}

String calories = intkcal_primer_min+"-"+intkcal_primer;
RecipeService service = retrofit.create(RecipeService.class);
Call<Respuesta> RespuestaCall = service.obtenerDatos(cantidad, q, app_id, app_key, health, calories);

RespuestaCall.enqueue(new Callback<Respuesta>() {
    @Override
    public void onResponse(Call<Respuesta> call, Response<Respuesta> response) {
        if (response.isSuccessful()) {
            Respuesta respuesta = response.body();
            ArrayList<Recipe> listaRecipe = respuesta.getHits();
            listaPrimeros = listaRecipe;
            COUNT_p = listaRecipe.size();
            obtenerSegundos( q: "main dish");//main dish

        } else {
            Log.e(TAG, msg: " onResponse: " + response.errorBody());
        }
    }

    @Override
    public void onFailure(Call<Respuesta> call, Throwable t) {
        Log.e(TAG, msg: " onFailure: " + t.getMessage());
    }
});

```

Figura 4.8: Código de conexión con la API EDAMAME

menú, aparte de las calorías, también tenemos que tener en cuenta el tipo de dieta y los alérgenos del usuario. Con esta información filtramos la búsqueda en la API, realmente se realizan dos búsquedas en esta, una para los primeros platos y otra para los segundos, almacenando estos en dos listas diferentes.

Para la implementación de dicha generación de menús se hace uso de un método recursivo, cada vez que se accede a este se generan dos números aleatorios, uno para obtener un primero y otro para obtener un segundo. Con estos dos platos se comprueba que la suma de sus macronutrientes (gramos, hidratos y proteínas) no superen la cantidad en gramos de ninguno de los macronutrientes que debe consumir el usuario, en caso de que alguno de ellos supere esta cuantía, se generarán otros dos platos aleatorios. Lo siguiente que se comprueba, es que no se repita exactamente ningún plato en la misma semana, ya que se dispone de una gran variedad de recetas. Posteriormente se comprueban los grupos alimenticios, verificando que no superen los márgenes establecidos en la Sección 2.1.3. En el caso de que estos se superen, se generarán otros dos platos aleatorios. La parte que más ralentiza la generación del menú, es la búsqueda del ingrediente principal ya que el diccionario de alimentos es muy grande, por ello se ha puesto al final, para que no tenga que hacer esta búsqueda muchas veces y así reducir al máximo la tardanza generando el menú. Una vez obtenido el ingrediente principal,

se comprueba que el del día anterior no haya sido el mismo, es decir que si ayer comí macarrones hoy no los coma otra vez. Por último, se almacenan los platos en la base de datos con toda la información necesaria. Cabe destacar que en la API no hay postres especialmente saludables y debido a la insistencia por parte de la OMS [21] del aumento del consumo de frutas, se ha decidido que el postre de todos los días de la semana será fruta, dejando la pieza de esta a elección del usuario, esto es debido a que dependiendo de la temporada hay determinadas frutas que no se dan y el valor calórico que estas tienen es muy parecido, por lo que se realizó una estimación de kilocalorías y esta se resta al total calórico que debe consumir el usuario en el almuerzo.

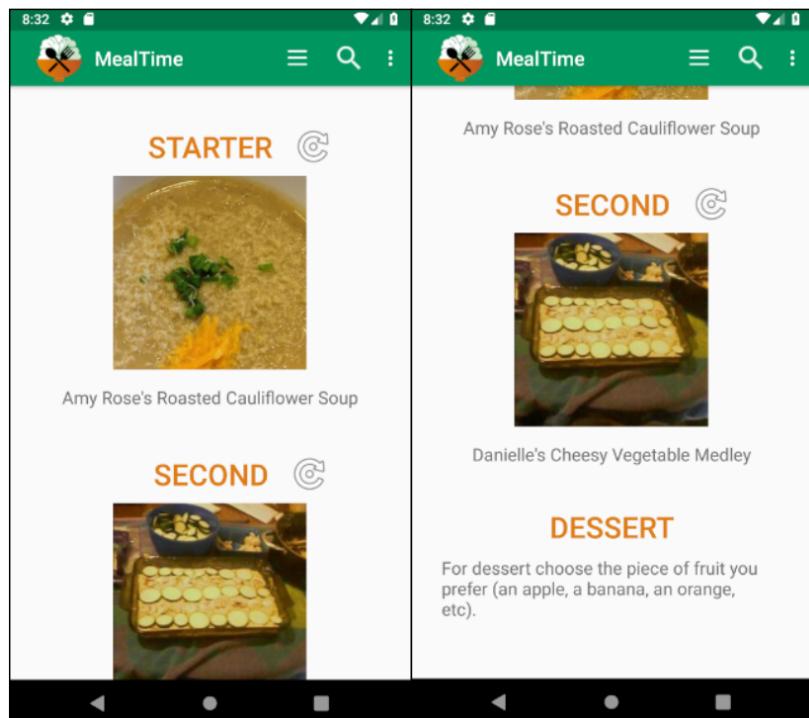


Figura 4.9: Pantalla del almuerzo de un día de la semana

En la pantalla del menú semanal (Figura 4.6), si se pulsa sobre cualquier día de la semana, se podrá acceder a los platos que componen el almuerzo de ese día (Figura 4.9). En esa pantalla se muestra una imagen ampliada del plato, con su nombre y al lado un botón que permite regenerar otro plato de forma aleatoria (siempre ajustándose a los márgenes de los distintos grupos alimenticios, al rango de calorías que debe consumir el usuario, así como las grasas, hidratos y proteínas de este, al igual que al tipo de dieta que siga o tipo de alérgenos que tenga).

Pulsando sobre alguno de los platos, se puede acceder a este (Figura 4.10), ahí se muestra más información del plato, como todos los ingredientes necesarios para su elaboración con los gramos especificados entre paréntesis, las calorías, los macronutrientes (proteínas, carbohidratos y grasas) que este tiene. También dispone en la parte inferior de dos botones, uno para la preparación del plato y otro para ver la tabla nutricional del mismo.

Al pulsar sobre el primer botón “How to make it”, se accede desde dentro de la aplicación al enlace de la preparación del plato, donde se explica paso a paso su elaboración (Figura 4.11).

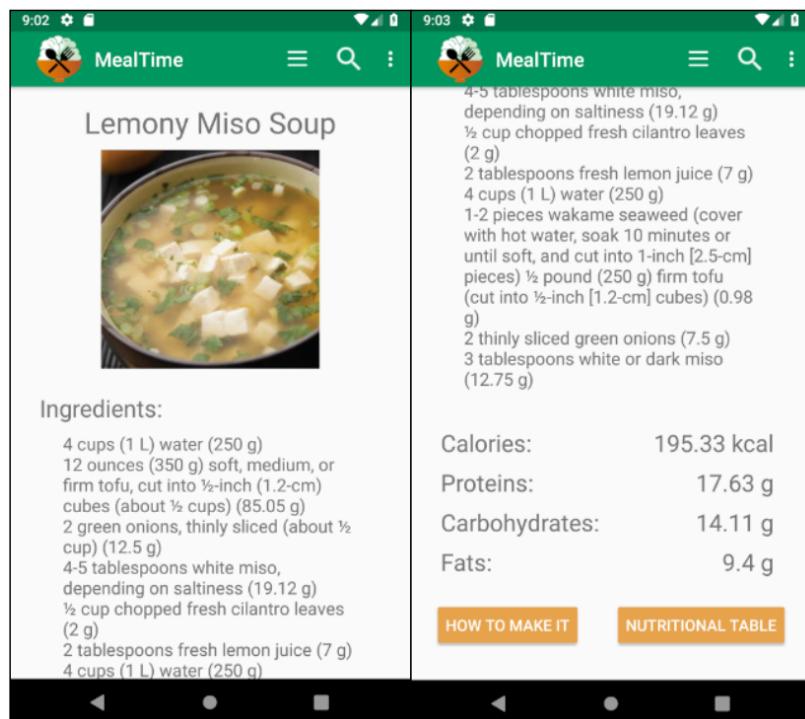


Figura 4.10: Pantalla de información del plato seleccionado



Figura 4.11: Pantalla de preparación del plato

Si en vez de pulsar el primer botón se pulsa el segundo “*Nutritional table*”, se accede desde dentro de la aplicación a un enlace que proporciona la API EDAMAM con una tabla nutricional muy completa del plato (Figura 4.12).

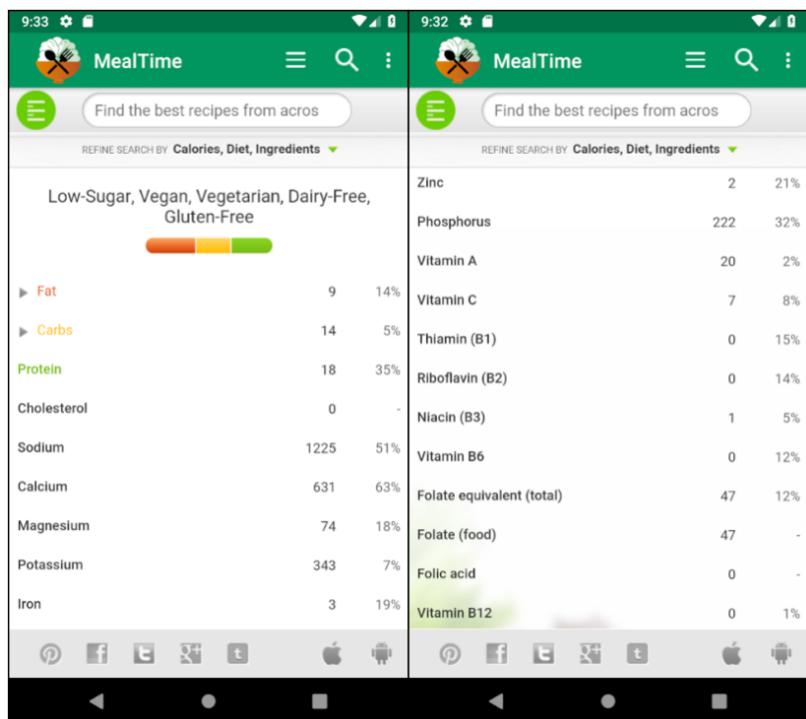


Figura 4.12: Pantalla de la tabla nutricional del plato

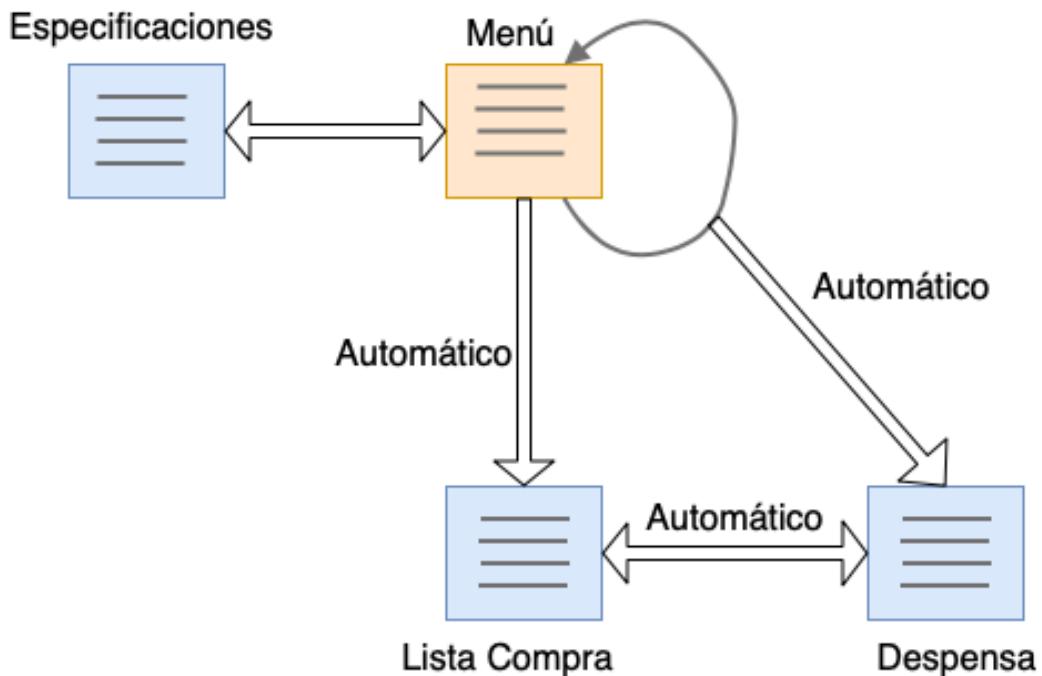


Figura 4.13: Esquema del funcionamiento de MealTime

#### 4.4. Gestión de la compra y la despensa

Para entender mejor el funcionamiento de la aplicación y cómo está esta interconectada, se proporciona el esquema que se muestra en la Figura 4.13. Tanto la información solicitada del usuario (especificaciones) como la generación del menú semanal ya se han explicado, sin embargo, las interconexiones entre el menú, la lista de la compra y la despensa puede que se entiendan mejor de forma visual haciendo uso del esquema nombrado. En cuanto a la despensa se pretende que el usuario tenga un lugar donde

añadir los alimentos de los que dispone junto con sus cantidades y que estos se borren automáticamente cuando se gasten. Con respecto a la lista de la compra, se mostrarán todos los ingredientes para elaborar los distintos platos semanales, al igual que la cantidad en gramos de cada uno de ellos. Al indicar que un ingrediente se ha comprado, este debe dejar de aparecer en la lista de la compra y añadirse a la despensa. Cada ingrediente de la lista de la compra tendrá un botón, este se usará para poder indicar que ese alimento ya lo tenemos en la despensa o para añadirlo a esta directamente.

Desde la pantalla de inicio (Figura 4.5), si se pulsa sobre el botón “Pantry”, se accederá a la despensa (Figura 4.14). Con el botón “Add” se podrá añadir un alimento nuevo o si se pulsa sobre uno ya existente, este podrá ser modificado o borrado.

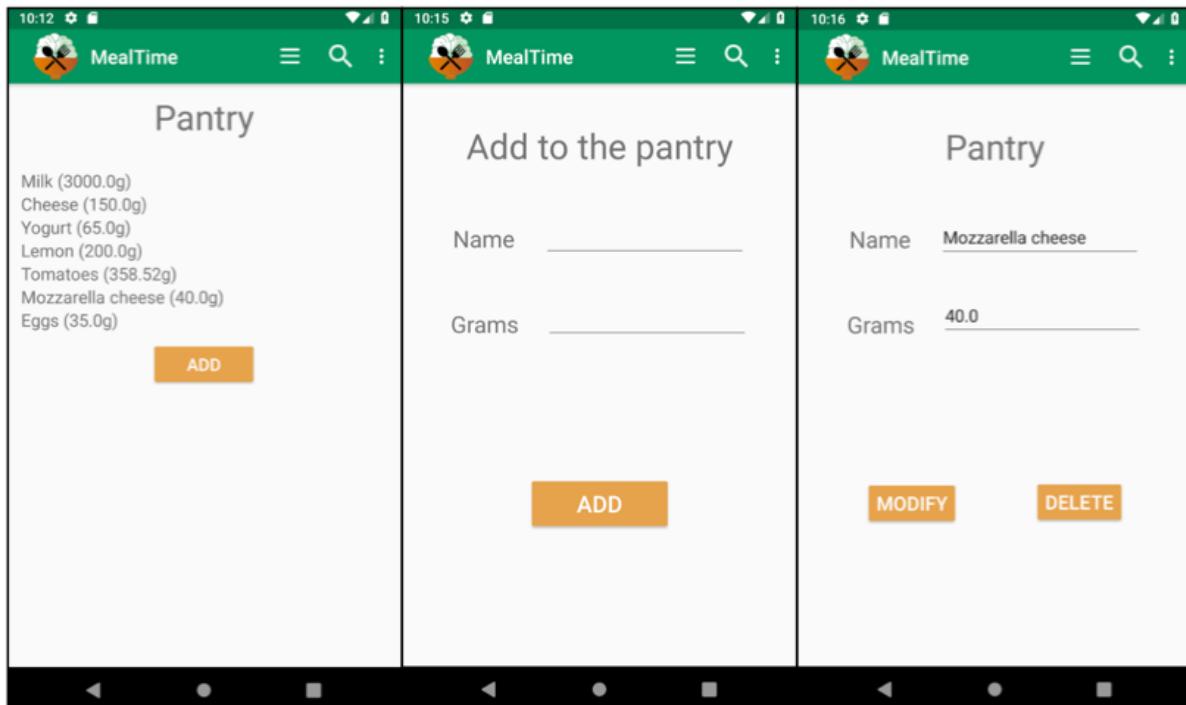


Figura 4.14: Pantalla de la despensa del usuario

Pulsando el botón “*Shopping list*”, se accede a la lista de la compra semanal. Al lado de cada ingrediente se especifica en gramos la cantidad correspondiente para una persona, ya que algunas recetas son para más de una. Si se selecciona la casilla que está junto al ingrediente, esta se marcará indicando que ya se ha comprado dicho ingrediente y la próxima vez que se entre en la lista de la compra, este ya no aparecerá, añadiéndose automáticamente en la despensa (Figura 4.15).

También puede darse el caso de que un ingrediente ya se tenga en la despensa, por lo que si se pulsa sobre el botón naranja que está al lado derecho del ingrediente, se accede a la despensa y se puede pulsar ese elemento que ya se tiene. Si las cantidades cuadran, es decir, si en la despensa se tiene mayor o igual cantidad que la que se necesita, entonces ese elemento se quita de la lista de la compra (Figura 4.16), pero en caso de no ser así se descuenta al ingrediente la cantidad que se tiene en la despensa y se mantiene en la lista de la compra con esta nueva cuantía (Figura 4.17).

Como la cantidad de ingredientes que se tendrá la despensa es de tamaño indefinido, ya que esta irá variando, para su implementación se hizo uso de un *RecyclerView*, con

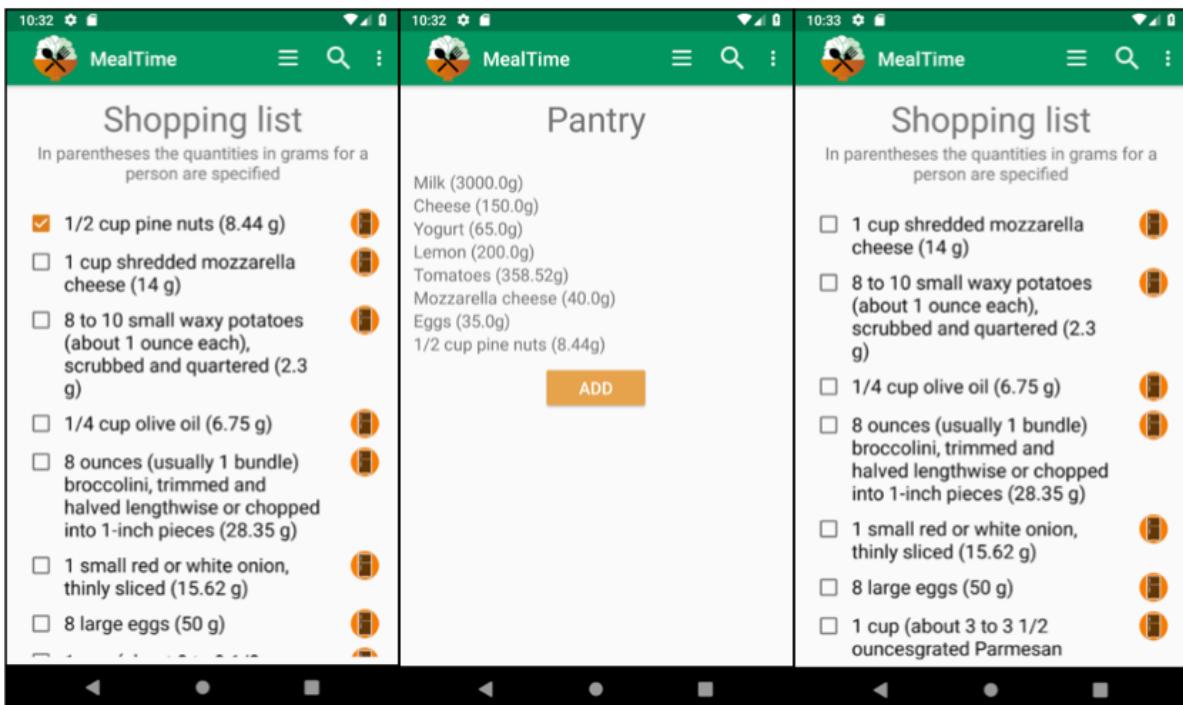


Figura 4.15: Pantalla de la lista de la compra

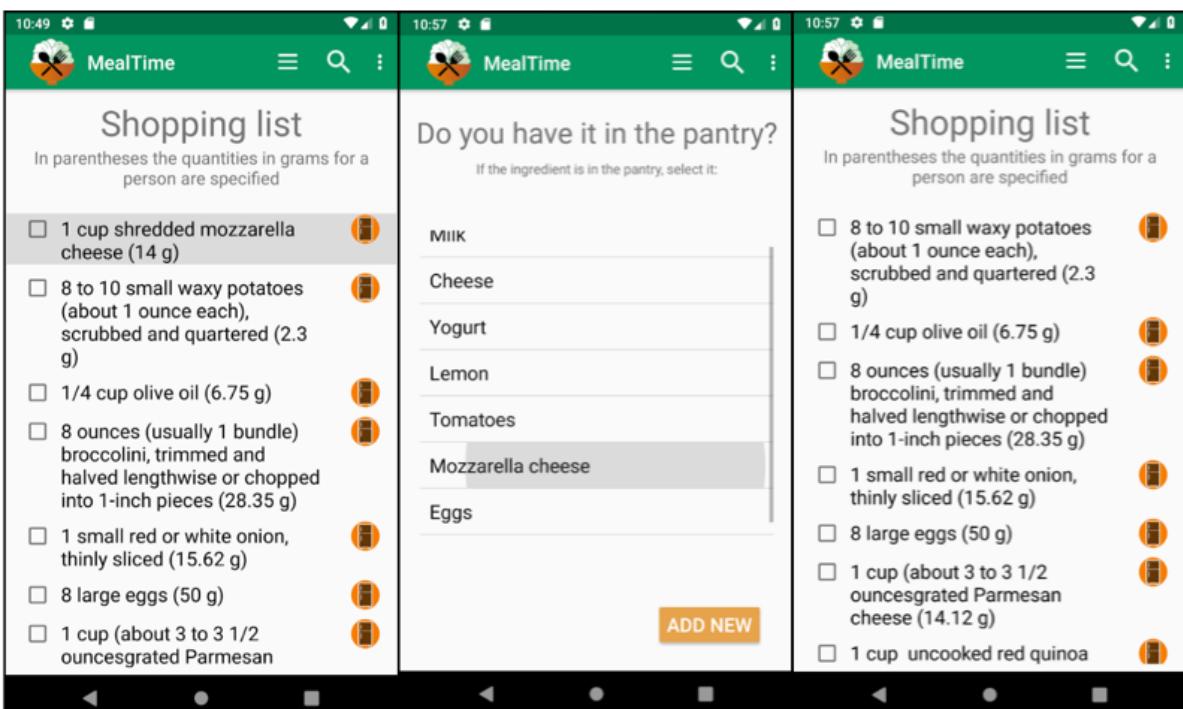


Figura 4.16: Ingrediente que se encuentra en la despensa

esto se consigue mostrar el contenido de listas de distintos tamaños. Para poder usarlo es necesario crear una clase *Adapter* (Figura 4.18) que realice la conexión de cada elemento de la lista con una posición en el *RecyclerView*. Para poder seleccionar un ingrediente de la lista y que se identifique este para realizar una acción, hice uso de la funcionalidad *setOnItemClickListener* que nos proporciona la clase anterior, con esto puedo indicar que al pulsar un elemento de la lista me envíe a otra pantalla para modificar la despensa con los datos del ingrediente seleccionado (Figura 4.19).

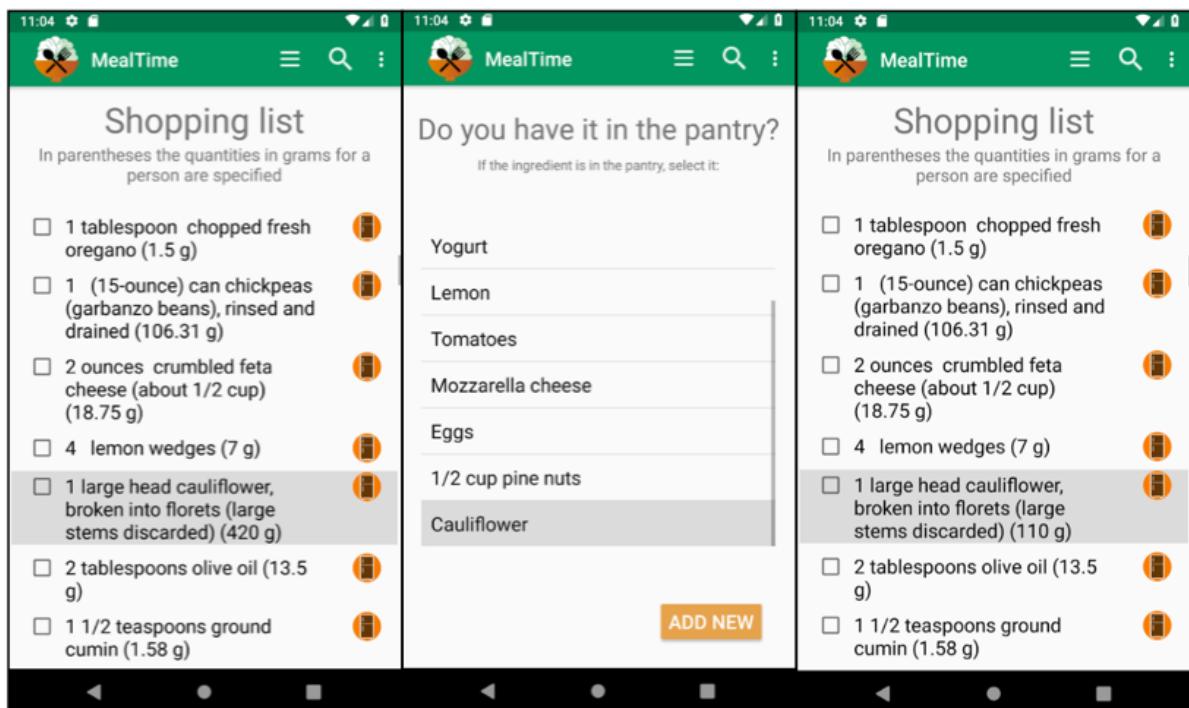


Figura 4.17: Ingrediente que se encuentra en la despensa con poca cantidad

```

public class AdapterDespensa extends RecyclerView.Adapter<AdapterDespensa.ViewHolderDatos>
    implements View.OnClickListener {
    ArrayList<Ingrediente_gramos> listDatos;
    private View.OnClickListener listener;

    public AdapterDespensa(ArrayList<Ingrediente_gramos> listDatos) {
        this.listDatos = listDatos;
    }

    @NonNull
    @Override
    public ViewHolderDatos onCreateViewHolder(@NonNull ViewGroup parent, int i) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_despensa, root: null, attachToRoot: false);
        view.setOnClickListener(this);
        return new ViewHolderDatos(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolderDatos holder, int position) {
        holder.asignarDatos( s: listDatos.get(position).getIngrediente() + " ("+listDatos.get(position).getGramos()+"g)");
    }

    @Override
    public int getItemCount() {
        return listDatos.size();
    }

    public void setOnClickListener(View.OnClickListener listener){
        this.listener=listener;
    }

    @Override
    public void onClick(View view) {
        if (listener!=null){
            listener.onClick(view);
        }
    }
}

public class ViewHolderDatos extends RecyclerView.ViewHolder {
    TextView dato;
    public ViewHolderDatos(@NonNull View itemView) {
        super(itemView);
        dato= (TextView) itemView.findViewById(R.id.itemDespensa);
    }

    public void asignarDatos(String s){ dato.setText(s); }
}
    
```

Figura 4.18: Clase Adapter

```

final ArrayList<Ingrediente_gramos> finalListDatos = listDatos;
adapter.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent siguiente = new Intent(getApplicationContext(), ModificarDespensa.class);
        siguiente.putExtra( name: "ingrediente", finalListDatos.get(recyclerDespensa.getChildAdapterPosition(view)).getIngrediente());
        siguiente.putExtra( name: "gramos", value: ""+finalListDatos.get(recyclerDespensa.getChildAdapterPosition(view)).getGramos());
        siguiente.putExtra( name: "id", value: ""+finalListDatos.get(recyclerDespensa.getChildAdapterPosition(view)).getId());
        startActivity(siguiente);
    }
});

```

Figura 4.19: Método onClick

La lista de la compra tiene dos acciones que realizar según donde se pulse, una es que al seleccionar en la casilla de la izquierda, esta se marca y se quite de la lista y la otra es que al pulsar sobre el botón de la derecha, envíe al usuario a otra pantalla con la lista de la despensa. Por ello la implementación de esta parte es más compleja que la implementación de la despensa, sin embargo, utiliza un método muy parecido. En este caso se usa un *ListView* para representar toda la lista de ingredientes, al igual que con la despensa, se hizo uso de un *Adapter* para realizar la conexión de cada elemento de la lista con una posición en el *ListView*. Dentro del *Adapter* se implementó un método *onClick*, igual que se hizo con la despensa (Figura 4.18), solo que en este caso lo que hace es controlar si se ha marcado la casilla que se encuentra al lado izquierdo del ingrediente, en caso de ser así, accede a la base de datos y realiza los cambios pertinentes para quitar el ingrediente de la lista de la compra y añadirlo a la despensa. Para controlar el botón que se encuentra en el lado derecho del ingrediente, se hizo de forma parecida que en la despensa, solo que en vez de usar el *setOnClickListener*, se hizo uso de la funcionalidad *setOnItemClickListener* y con esto se consiguió que al pulsar sobre dicho botón se pueda acceder a otra pantalla, de la misma forma que se hizo con la despensa (Figura 4.19).

## 4.5. Búsquedas de recetas e información nutricional

Desde la pantalla de inicio (Figura 4.5), si se pulsa el botón “Search recipe”, se accede a una nueva pantalla en la que se pueden buscar numerosas recetas (Figura 4.20). A esta pantalla también se puede acceder desde cualquier parte de la aplicación pulsando sobre la lupa que se encuentra en la parte superior derecha, junto al menú desplegable nombrado con anterioridad. Junto a la lupa tenemos tres rallas horizontales y pulsando sobre estas se podrá acceder a la pantalla de inicio (Figura 4.5).

Para la implementación de la búsqueda de recetas se estableció la conexión con la API al igual que se hizo en la Sección 4.3, con el cliente *REST Retrofit*. Esta vez no se almacena una lista de primeros o de segundos, se almacena una lista de platos filtrados por la o las palabras que introduzca el usuario. Al igual que se hizo con la despensa, la pantalla de búsqueda también hace uso de un *RecyclerView* con su correspondiente *Adapter* y *setOnClickListener* de forma similar a como se implementó en dicha despensa. Cabe destacar que ahora cada elemento de la lista, en vez de ser solo una cadena de texto, se compone de una imagen y una cadena de texto con el nombre de la receta.

Esta pantalla da libertad al usuario de buscar cualquier plato ya sea sano o no, siempre que no se pase de su límite calórico y teniendo en cuenta el tipo de dieta que siga o si posee algún alérgeno.

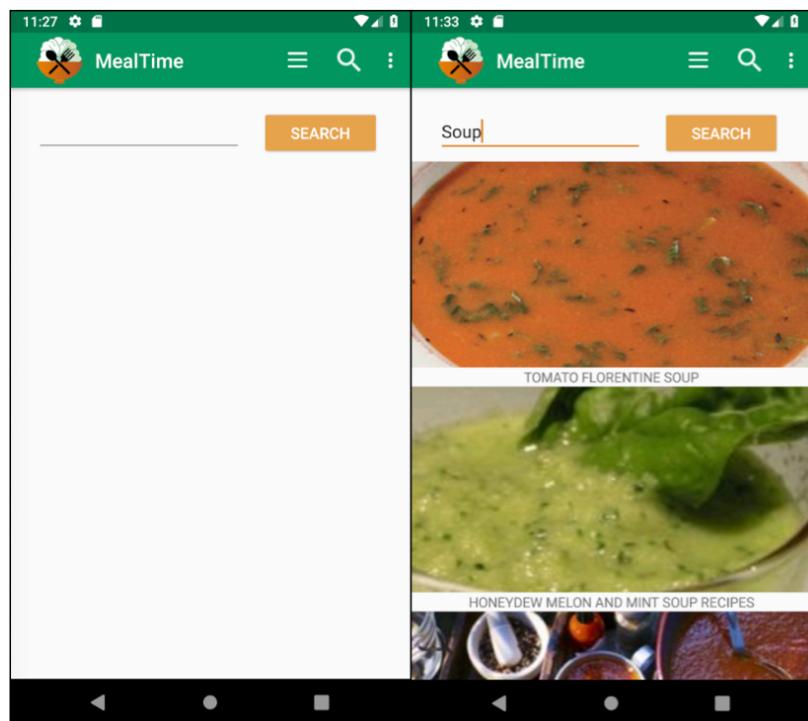


Figura 4.20: Pantalla de búsqueda

Volviendo a la pantalla de inicio de la aplicación (Figura 4.5), si se pulsa sobre el botón “Check calories” se puede observar la cantidad diaria que el usuario debe consumir de calorías, grasas, proteínas e hidratos (Figura 4.21). A esta pantalla también se puede acceder desde el menú desplegable que se encuentra en la parte superior derecha de la aplicación (Figura 4.4).

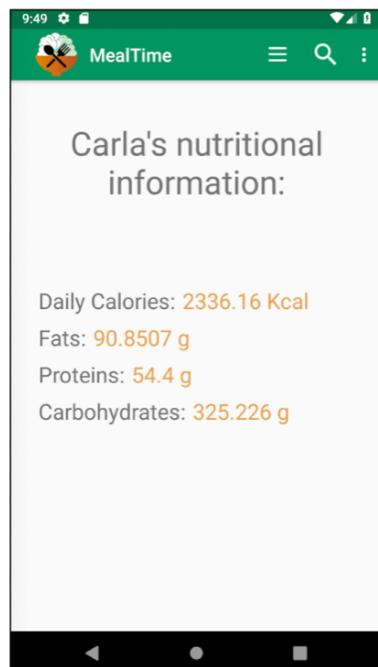


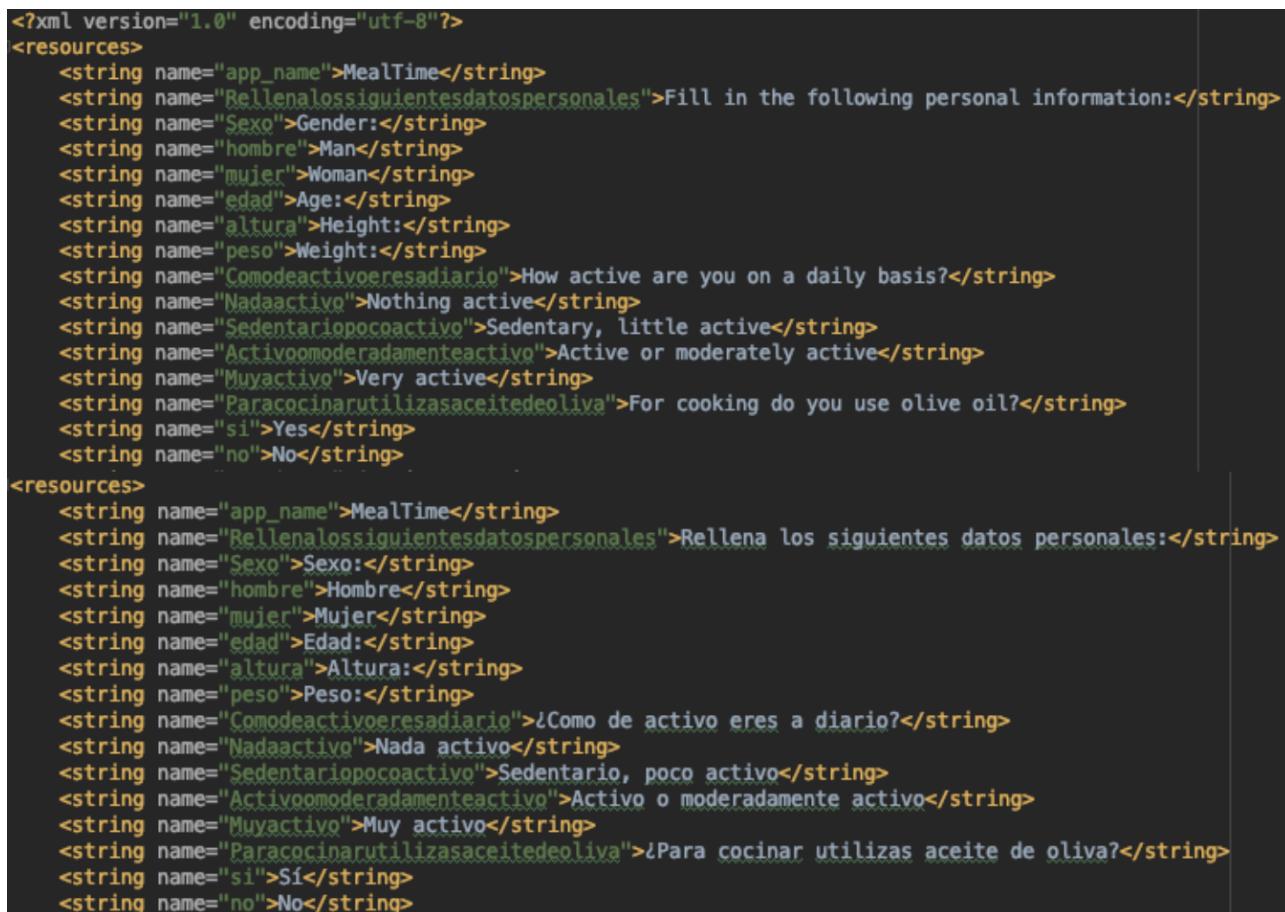
Figura 4.21: Pantalla de información nutricional del usuario

Con los datos que se solicitan del usuario y haciendo uso de los cálculos mostrados en el Capítulo 2, implementé el código necesario para realizar el cómputo de kilocalorías,

grasas, proteínas e hidratos que debe consumir el usuario al día.

## 4.6. Idioma

Como ya comentamos con anterioridad, la API EDAMAM proporciona las recetas en dos idiomas, español e inglés. Se ha decidido usar este último para la aplicación MealTime, ya que la versión en español de la API se encuentra en beta, aun así, establecí la conexión con esta última para ver cómo funcionaba, ya que en el JSON que proporcionaba, se veían todos los datos y aparentemente daba una respuesta correcta, pero al intentar acceder a las fotos desde la aplicación no se mostraban, supongo que es debido a que esta versión se encuentra en desarrollo.



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">MealTime</string>
    <string name="Rellenalossiguientesdatospersonales">Fill in the following personal information:</string>
    <string name="Sexo">Gender:</string>
    <string name="hombre">Man</string>
    <string name="mujer">Woman</string>
    <string name="edad">Age:</string>
    <string name="altura">Height:</string>
    <string name="peso">Weight:</string>
    <string name="Comodeactivoeresadiario">How active are you on a daily basis?</string>
    <string name="Nadaactivo">Nothing active</string>
    <string name="Sedentariopocoactivo">Sedentary, little active</string>
    <string name="Activoomoderadamenteactivo">Active or moderately active</string>
    <string name="Muyactivo">Very active</string>
    <string name="Paracocinarutilizasaceitedeoliva">For cooking do you use olive oil?</string>
    <string name="si">Yes</string>
    <string name="no">No</string>
<resources>
    <string name="app_name">MealTime</string>
    <string name="Rellenalossiguientesdatospersonales">Rellena los siguientes datos personales:</string>
    <string name="Sexo">Sexo:</string>
    <string name="hombre">Hombre</string>
    <string name="mujer">Mujer</string>
    <string name="edad">Edad:</string>
    <string name="altura">Altura:</string>
    <string name="peso">Peso:</string>
    <string name="Comodeactivoeresadiario">¿Como de activo eres a diario?</string>
    <string name="Nadaactivo">Nada activo</string>
    <string name="Sedentariopocoactivo">Sedentario, poco activo</string>
    <string name="Activoomoderadamenteactivo">Activo o moderadamente activo</string>
    <string name="Muyactivo">Muy activo</string>
    <string name="Paracocinarutilizasaceitedeoliva">¿Para cocinar utilizas aceite de oliva?</string>
    <string name="si">Sí</string>
    <string name="no">No</string>
```

Figura 4.22: Fichero strings.xml

Actualmente la aplicación está en inglés, pero también dispone de una traducción al español para si en un futuro terminan la beta de EDAMAM, que se pueda usar esta también en español. La aplicación no está restringida solo al inglés o al español ya que esta se puede traducir a cualquier idioma con mucha facilidad. Esto es así porque todo texto que aparece, se encuentra en un fichero llamado `strings.xml` (Figura 4.22) y para poder usar otro idioma, simplemente hay que seleccionar el nuevo idioma y realizar la traducción (Figura 4.23). Esto es gracias a una clave que le he puesto a cada texto, se refencian automáticamente y según el idioma que tenga el usuario en su dispositivo móvil, verá la aplicación en dicho idioma.

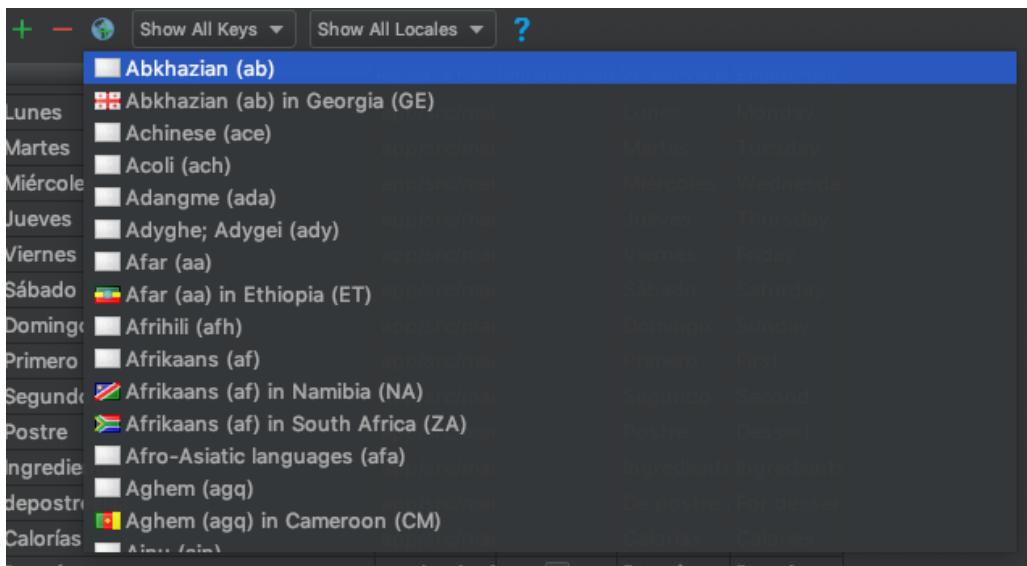


Figura 4.23: Selección de un nuevo idioma

## 4.7. Repositorio

Para el desarrollo del proyecto es interesante disponer de un historial de cambios realizados, asegurando así que en caso de fallo siempre se puede volver a una versión anterior. Es por ello que durante la implementación de la aplicación se hizo uso del control de versiones **Git** por medio de la plataforma **GitHub**, realizando un **commit** por cada implementación lograda y luego haciendo un **push** al repositorio remoto.

Todo el código implementado se encuentra accesible desde el repositorio en **GitHub**:

<https://github.com/ULL-ESIT-INF-TFG-1819/ull-esit-inf-tfg-1819-menu-planning-android-alu0100768152.git>

# **Capítulo 5**

## **Conclusiones y líneas de trabajo futuras**

### **5.1. Conclusiones**

Los malos hábitos de vida se han convertido en la mayor amenaza para la salud mundial, especialmente las dietas desequilibradas. Muchas personas asocian la mala alimentación con un problema estético de subir de peso, pero este no es el mayor problema, una malnutrición supone un riesgo grave en la salud del individuo dando lugar a un aumento considerable de diabetes, sobrepeso y obesidad. Poco a poco la sociedad se va concienciando más de la importancia de tener una buena alimentación y que esta sea equilibrada. En la búsqueda que realicé de aplicaciones y herramientas existentes, no encontré ninguna que recogiera todas las características necesarias para ayudar a mejorar el problema de la malnutrición tal y como lo hace MealTime.

Los principales objetivos marcados se han cumplido, estos eran la generación de un menú semanal personalizado teniendo en cuenta las características fisiológicas del usuario, que esta generación del menú se hiciera por medio de una aplicación móvil, y que esta proporcionara tanto una lista de la compra, como una despensa que estén interconectadas entre sí. En un principio, se propusieron una serie de objetivos los cuales no se pudieron lograr, tal vez por lo ambiciosos que estos eran. Uno de esos objetivos fue proporcionar un coste a cada plato, pero no conseguimos encontrar una API o base de datos que, aparte de todas las características que necesitábamos para generar el menú, también nos proporcionara un precio y, teniendo en cuenta la problemática de que los ingredientes de la API EDAMAM se encontraban en una cadena de texto junto con más información, decidimos descartar esta idea. También quisimos que el menú no fuera solo semanal, sino que el usuario pudiera decidir también hacerlo mensual, pero debido al tiempo que llevó hacer un menú semanal y a las dificultades que este llevaba, consideramos solo hacerlo semanal, ya que, si hubiéramos hecho el mensual también, seguramente la aplicación no tendría ni lista de la compra ni despensa por la falta de tiempo.

La idea de crear una aplicación móvil me entusiasmaba muchísimo, pero no era consciente de lo difícil que era implementarla, ya que esta posee muchas funcionalidades que deben estar conectadas unas con otras, ahora se realmente como se fabrica una aplicación y todo el trabajo que hay detrás, aun así, me siento orgulloso de decir que es el proyecto más grande que he realizado hasta el momento y me encantaría seguir por esta

línea de trabajos en un futuro.

## 5.2. Líneas de trabajo futuro

En cuanto al trabajo futuro de la aplicación *MealTime*, se le podrían realizar ciertas mejoras y añadirle algunas funcionalidades que hagan de esta, una aplicación más completa de lo que es.

A continuación, se nombrarán una serie de propuestas:

- Como se ha comentado en la Sección 4.6, la aplicación está prácticamente preparada para la versión en español, simplemente habría que esperar a que se finalice la beta de la API EDAMAM y cambiar en esta versión la URL raíz por la que proporcionen para este idioma.
- Sería interesante dotar a la aplicación de un menú mensual y que el propio usuario, pudiera decidir si quiere generar uno semanal o mensual, controlando en la lista de la compra, si esta se quiere hacer cada semana o cada cierta cantidad de tiempo.
- Aunque no conseguimos encontrar una herramienta que nos proporcione precios de los distintos alimentos, ni la forma de enlazar estos con la API, sería muy interesante proporcionar un precio estimado de lo que costaría cada plato, por lo que si en un futuro se encuentra una API o herramienta que proporcione precios en tiempo real de distintos alimentos, sería interesante acoplarlo a la aplicación *MealTime* para ofrecer ese dato al usuario.
- Una mejora interesante en la aplicación, sería que se ofrecieran recomendaciones nutricionales para el resto de comidas del día, según el menú generado y los márgenes de los distintos grupos alimenticios.
- También se podrían generar tanto cenas como desayunos en el menú, ajustando al correspondiente porcentaje calórico, controlando que al día no supere la cantidad de macronutrientes, los márgenes de los distintos grupos alimenticios etc.

# **Capítulo 6**

## **Conclusions and Future work**

### **6.1. Conclusions**

Bad lifestyle habits have become the biggest threat to global health, especially unbalanced diets. Many people associate poor diet with an aesthetic problem of gaining weight, but this is not the biggest problem, malnutrition is a serious risk to the health of people producing a considerable increase in diabetes, overweight and obesity. Little by little society is becoming more aware of the importance of having a good diet and that this is balanced. In the search, I did of existing applications and tools, I did not find any that had all the necessary characteristics to help improve the problem of malnutrition as MealTime does.

The main objectives have been met, these were the generation of a personalized weekly menu considering the physiological characteristics of the user, that this generation of the menu was made through a mobile application, and that this provide a shopping list and a pantry that are interconnected with each other. At the beginning, my tutor and I set ourselves many objectives which could not be achieved, maybe because of the ambitious that they were, one of those objectives was to provide a cost to each dish, but we could not find an API or database that, apart from all the features we needed to create the menu, it also provides us with a price and, as the ingredients of the EDAMAM API had more information next to the ingredient, we decided not to do this functionality. We also wanted that the menu was not only weekly, that the user could also decide to do it monthly, but due to the time it took to make a weekly menu and the difficulties that it had, we decided to only do it weekly.

The idea of creating a mobile application excited me a lot, but I didn't know how difficult it was to implement it, since it has many features that must relate to each other, now i really know how an application is manufactured and all the work behind it, even so, I am proud to say that it is the biggest project I have done so far and I would love to continue with this type of work in the future.

### **6.2. Future work**

For the future work of the MealTime application, could be make certain improvements and add some features that make this application more complete than it is, hereunder, several proposals will be named:

- As mentioned in Section 4.6, the application is practically ready for the Spanish version, we just must wait until the EDAMAM API beta is finished and change in this version the root URL by which they provide for this language.
- It would be interesting to give the application a monthly menu and the user would decide if he wants to generate a weekly or monthly menu, and the same for the shopping list.
- Although we did not find a tool to provide prices for different foods, nor the way to link these with the API, it would be very interesting to provide an estimated price of what each plate would cost, so if in the future there is an API or tool that provides real-time prices of different foods, it would be interesting to attach it to the MealTime application to offer that information to the user.
- An interesting improvement in the application would be to offer nutritional recommendations for the rest of the meals of the day, according to the menu generated and the margins of the different food groups.
- Both dinners and breakfasts could be generated in the menu, adjusting to the corresponding caloric percentage, controlling that the daily quantity does not exceed the quantity of macronutrients, the margins of the different food groups, etc.

# Capítulo 7

## Presupuesto

Para poder realizar un coste total, debemos tener en cuenta las distintas tareas realizadas al igual que las herramientas implicadas en estas y su respectivo coste. Las principales tareas realizadas fueron las siguientes:

- **Análisis del estado del arte**, esto consistió en analizar las aplicaciones y herramientas existentes en el ámbito de la dietética y nutrición, sobre todo las que incluyen planificación automática de menús, y realizar una comparativa de estas.
- **Análisis de tecnologías para el desarrollo de aplicaciones móviles**, se hizo una búsqueda de las distintas tecnologías y herramientas que hay para el desarrollo de aplicaciones en Android, se buscó información sobre estas y por último me decanté por una.
- **Búsqueda de bases de datos de alimentos**, esto consistió en analizar distintas bases de datos y APIs que proporcionaron recetas con las características que necesitábamos para la generación de menús. La API seleccionada fue EDAMAM, esta tiene una versión gratuita, pero está limitada por un número de consultas mensuales, si se quiere hacer un uso intensivo de esta se debe pagar una cuota de 25,65 euros al mes.
- **Información sobre el tema en cuestión**, en este caso la nutrición, para ello se consultaron distintos libros [35], principalmente me guie del Libro Blanco de la Nutrición en España [31] y también se consultó mucha información publicada por la OMS [21].
- **Diseño del logo**, se generó una imagen que representa la esencia de la aplicación, para que esta aparezca como ícono de dicha aplicación.
- **Diseño e implementación de la herramienta**, esto es, todo el desarrollo de la aplicación, tanto el diseño, como todas las funcionalidades de la misma.
- **Documentación**, consistió en la elaboración de la memoria de este trabajo, haciendo uso de Latex [26] por medio de OverLeaf [12].

En primer lugar, nombraré las distintas tecnologías usadas y el coste de cada una de ellas:

Tecnología	Precio
Github	0€
API EDAMAM	25,65€ al mes
Android Studio	0€
Java	0€
Latex y OverLeaf	0€
SQLite	0€

Tabla 7.1: Presupuesto de las tecnologías usadas

En la Tabla 7.2 se estimará el coste de la realización de cada una de las tareas anteriormente nombradas y, finalmente, el coste total del proyecto.

Tareas	Horas empleadas	Precio
Análisis del estado del arte	30 horas	300€ (10€/h)
Análisis de tecnologías para el desarrollo de aplicaciones móviles	15 horas	150€ (10€/h)
Búsqueda de bases de datos de alimentos	40 horas	480€ (12€/h)
Información sobre el tema en cuestión	20 horas	240€ (12€/h)
Diseño del logo	4 horas	90€
Diseño e implementación de la herramienta	170 horas	2.550€ (15€/h)
Documentación	40 horas	400€ (10€/h)
<b>TOTAL</b>	<b>319 horas</b>	<b>4.210€</b>

Tabla 7.2: Presupuesto del coste por tarea

A este coste total habría que añadirles el precio de la API EDAMAM, teniendo en cuenta que se usó durante 4 meses, el **coste total** del proyecto sería **4.312,6€**.

# Bibliografía

- [1] Android studio. <https://developer.android.com/studio>. Accedido 20-03-2019.
- [2] Appcelerator titanium. <https://www.appcelerator.com/>. Accedido 01-04-2019.
- [3] Basic4android. <https://www.b4x.com/>. Accedido 09-04-2019.
- [4] Css. <https://www.w3schools.com/css/default.asp>. Accedido 08-04-2019.
- [5] Dietopro. <https://dietopro.com/>. Accedido 02-03-2019.
- [6] Html5. <https://developer.mozilla.org/es/docs/HTML/HTML5>. Accedido 08-04-2019.
- [7] Javascript. <https://www.javascript.com/>. Accedido 25-03-2019.
- [8] Kotlin. <https://kotlinlang.org/>. Accedido 09-04-2019.
- [9] Livecode. <https://livecode.com/>. Accedido 10-04-2019.
- [10] Mono for android. <https://www.monodevelop.com/archived/download/mono-for-android/>. Accedido 10-04-2019.
- [11] Nutritionix. <https://www.nutritionix.com/>. Accedido 06-03-2019.
- [12] Overleaf. <https://es.overleaf.com/>. Accedido 05-06-2019.
- [13] Phonegap. <https://phonegap.com/>. Accedido 06-04-2019.
- [14] Ruboto. <http://ruboto.org/>. Accedido 10-04-2019.
- [15] Bedca. <http://bedca.net/>, 2006. Accedido 04-03-2019.
- [16] Kowilimón. <https://dietopro.com/>, 2013. Accedido 14-03-2019.
- [17] Cocina para hombres. <http://www.cocinaparahombres.com/recetas/receta.php?rec=lomo-naidad>, 2019. Accedido 04-03-2019.
- [18] Edamam. <https://developer.edamam.com/edamam-recipe-api>, 2019. Accedido 18-05-2019.
- [19] Menú planner. <https://nestlemenuplanner.es/menu-semanal-366.html>, 2019. Accedido 04-03-2019.
- [20] Objetivo bienestar. [https://www.objetivobienestar.com/menu-semanal-saludable\\_115\\_119.html](https://www.objetivobienestar.com/menu-semanal-saludable_115_119.html), 2019. Accedido 02-03-2019.

- [21] Organización mundial de la salud. <https://www.who.int/es>, 2019. Accedido 07-04-2019.
- [22] Rapidapi. <https://rapidapi.com/>, 2019. Accedido 12-05-2019.
- [23] Recipe search api documentation. <https://developer.edamam.com/edamam-docs-recipe-api>, 2019. Accedido 20-5-2019.
- [24] Webknox recipe api. <https://rapidapi.com/webknox/api/recipe>, 2019. Accedido 10-03-2019.
- [25] ¿qué hay en la nevera? <http://www.quehayenlanevera.com/nuevo-menu>, 2019. Accedido 04-03-2019.
- [26] W. M. F. Alexánder Borbón A. *Edición de Textos Científicos Latex*. Instituto Tecnológico de Costa Rica, 2017.
- [27] V. Autores. Retrofit. <https://square.github.io/retrofit/>. Accedido 19-05-2019.
- [28] L. G. de Ernesto. Curso android desde cero con android studio. <https://www.youtube.com/watch?v=tyx05coXixw&list=PLyvsggKtwbLX06iMtXnRGX5lyjiiMaT2y&index=1>, 2017. Accedido 24-3-2019.
- [29] J. M. L. de Guevara. *Fundamentos de programación en Java*. EME.
- [30] U. P. de Valencia. Android: Introducción a la programación. <https://www.edx.org/es/course/android-introduccion-a-la-programacion-2>, 2014. Accedido 08-02-2019.
- [31] G. V. Moreiras. *Libro Blanco de La Nutrición es España*. Fundación Española de la Nutrición (FEN), 2013.
- [32] J. M. R. Pérez. Técnicas de optimización multi-objetivo aplicadas al problema de planificación de menús. <https://riull.ull.es/xmlui/handle/915/2611>, 2016.
- [33] D. Robledo. *Desarrollo de aplicaciones para Android I*.
- [34] R. Whitcomb. Food json library. <https://think.cs.vt.edu/corgis/json/food/food.html>, 06/21/2016. Accedido 22-05-2019.
- [35] Ángeles Carbajal Azcona. *Manual de Nutrición y Dietética*. 2013.